

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD FFFFFFFFFFFFFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEEEEEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEEEEEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF

```

[illegible]

[illegible][illegible]

[IDENT ('V04-000'),

{ **

```
*****
*
*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*   ALL RIGHTS RESERVED.
*
*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*   TRANSFERRED.
*
*   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*   CORPORATION.
*
*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

FACILITY: VAX/VMS EDF (EDIT/FDL) UTILITY

ABSTRACT: This facility is used to create, modify, and optimize
FDL specification files.

ENVIRONMENT: NATIVE/USER MODE

AUTHOR: Ken F. Henderson Jr.

CREATION DATE: 27-Mar-1981

MODIFIED BY:

V03-010	RRB0009	Rowland R. Bradley	22 Jan 1984
		Enhancement for display of # buckets in index, # of Pages to cache index, and average # key exams.	
V03-009	KFH0009	Ken Henderson	10 Sep 1983
		Support for named UICs.	
V03-008	KFH0008	Ken Henderson	8 Aug 1983
		Changes for seperate compilation.	
V03-007	KFH0007	Ken Henderson	26 Apr 1983
		Added FOUND_0, PRIMARY_INDEX_BUCKETS. Added a bunch of initial values; to reduce the code in INIT_EDITOR. Added SEC_ATTR, NUMBER_KEYS_SET.	

EDFVAR
V04-000

Source Listing

J 6
16-Sep-1984 00:42:36
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277
DISK&VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (1)

Page 2

V03-006	KFH0006	Ken Henderson	14 Apr 1983
	Added SET FUNCTION, GRANULARITY, PROMPTING, RESPONSES, JOURNAL_FILE, JOURNAL_ENABLED, support for SEGMENTED keys.		
V03-005	KFH0005	Ken Henderson	31 Jan 1983
	Changed reference of FDL\$TYPE to FDL3\$TYPE.		
V03-004	KFH0004	Ken Henderson	20 Jan 1983
	Removed DASH, and added DEPTHPOINT_LEFT, _MID, _RIGHT.		
V03-003	KFH0003	Ken Henderson	8 Sept 1982
	Consolidated many main variables into the xDATA arrays.		
V03-002	KFH0002	Ken Henderson	2 April 1982
	Added EDF\$GL_SECNUM.		
V03-001	KFH0001	Ken Henderson	23-Mar-1982
	Took out EDITEDL.		

--)

```
0085 ENVIRONMENT ('LIB$EDFVAR'),
0086
0087 INHERIT (
0088
0089 'SYSS$LIBRARY:STARLET',
0090 'SHRLIB$FDLPARDEF',
0091 'LIB$EDFCNST',
0092 'LIB$EDFTYPE',
0093 'LIB$EDFSTRUCT',
0094
0095 )]
0096
0097 MODULE EDFVAR;
0098
0099 VAR
0100
0101 { +
0102 These are for the FDL file(s).
0103 - }
0104 FDL_DEST          : [VOLATILE] TEXT;
0105 TT                : TEXT;
0106 JOURNAL_FILE      : [VOLATILE] TEXT;
0107 JOURNAL_ENABLED   : [VOLATILE] BOOLEAN := FALSE;
0108 JOURNAL_FILENAME  : STRING255;
0109 INPUT_FILENAME_DESC : DESCRIPTOR;
0110 OUTPUT_FILENAME_DESC : DESCRIPTOR;
0111 ANALYSIS_FILENAME_DESC : DESCRIPTOR;
0112 RES_OUTPUT_FILENAME_DESC : [VOLATILE] DESCRIPTOR;
0113 DEFAULT_FILENAME_DESC : DESCRIPTOR;
0114 NL_DEV_DESC       : DESCRIPTOR;
0115 ANALYSIS_SPECIFIED : BOOLEAN := FALSE;
0116 ANALYSIS_ONLY     : [VOLATILE] BOOLEAN := FALSE;
0117
0118 { +
0119 These are the main editing control variables.
0120 While editing is true, we stay in the editor.
0121 Take defaults is true if we're answering questions
0122 automatically.
0123 If no input is true, we skip the input fdl file operation.
0124 If auto tune is true, we don't do any output to the terminal,
0125 or expect any from it. It's all an automated Optimize script.
0126 - }
0127 EDITING          : [VOLATILE] BOOLEAN := FALSE;
0128 TAKE_DEFAULTS    : BOOLEAN := FALSE;
0129 NO_INPUT         : BOOLEAN := FALSE;
0130 AUTO_TUNE        : [VOLATILE] BOOLEAN := FALSE;
0131
0132 { +
0133 This is for outputting FDL to the terminal.
0134 - }
0135 SYSS$OUTPUT_NAME : [VOLATILE] STRING11 := 'SYSS$OUTPUT: ';
0136
0137 { +
0138 These indicate whether classes of errors were detected.
0139 - }
0140 SYSS$INPUT_ERROR : [VOLATILE] BOOLEAN := FALSE;
0141 RMS_INPUT_ERROR  : [VOLATILE] BOOLEAN := FALSE;
```

```
0142 RMS_OUTPUT_ERROR : [VOLATILE] BOOLEAN := FALSE;
0143 CONTROL_ZEE_TYPED : [VOLATILE] BOOLEAN := FALSE;
0144 MAIN_CTRLZ : [VOLATILE] BOOLEAN := FALSE;
0145 MAIN_LEVEL : [VOLATILE] BOOLEAN := TRUE;
0146 QUESTION_TYPED : [VOLATILE] BOOLEAN := FALSE;
0147
0148 { +
0149 This is for graphing.
0150 - }
0151 XY_PLOT : XY_PLOT_TYPE;
0152 COLOR_PLOT : XY_PLOT_TYPE;
0153 XY_ARRAY : XY_ARRAY_TYPE;
0154 COLOR_ARRAY : XY_ARRAY_TYPE;
0155 COLOR_ROW : PACKED ARRAY [0..(BKT$C_MAXBKTSIZ-1)] OF BYTE;
0156 BREAKPOINT_LEFT : INTEGER;
0157 BREAKPOINT_MID : INTEGER;
0158 BREAKPOINT_RIGHT : INTEGER;
0159 DEPTHPOINT_LEFT : INTEGER;
0160 DEPTHPOINT_MID : INTEGER;
0161 DEPTHPOINT_RIGHT : INTEGER;
0162 EXAMPOINT_LEFT : INTEGER;
0163 EXAMPOINT_MID : INTEGER;
0164 EXAMPOINT_RIGHT : INTEGER;
0165 NUMPOINT_LEFT : INTEGER;
0166 NUMPOINT_MID : INTEGER;
0167 NUMPOINT_RIGHT : INTEGER;
0168 PAGEPOINT_LEFT : INTEGER;
0169 PAGEPOINT_MID : INTEGER;
0170 PAGEPOINT_RIGHT : INTEGER;
0171 GRAPH_TYPE : INTEGER;
0172 CURRENT_GRAPH_INDEX : INTEGER;
0173 LAST_GRAPH_INDEX : INTEGER;
0174 STEPS : INTEGER;
0175 Y_LABEL : STRING32;
0176
0177 { +
0178 These are the 'width' arrays that indicate how long a particular keyword
0179 should be typed.
0180 - }
0181 PRIMARY_WIDTH : PACKED ARRAY [PRIMARY_TYPE] OF BYTE;
0182 SECONDARY_WIDTH : PACKED ARRAY [SECONDARY_TYPE] OF BYTE;
0183
0184 { +
0185 This array stores the maximum value of the number-valued secondaries.
0186 - }
0187 SECONDARY_MAX : ARRAY [SECONDARY_TYPE] OF INTEGER;
0188
0189 { +
0190 This array stores the legal sequencing of Primaries as defined by the
0191 FDL Specification.
0192 - }
0193 PRI_SEQ : [VOLATILE] PACKED ARRAY [PRIMARY_TYPE] OF BYTE;
0194
0195 { +
0196 These store the character sequences to set the video attribute modes
0197 of the VT100 (and compatible) terminals.
0198 - }
```



```
0199 ANSI_RESET      : [VOLATILE] STRING4 := ('(27)', [' ', '0', 'm']);
0200 ANSI_BOLD       : STRING4 := ('(27)', [' ', '1', 'm']);
0201 ANSI_UNDERSCORE : STRING4 := ('(27)', [' ', '4', 'm']);
0202 ANSI_BLINK      : STRING4 := ('(27)', [' ', '5', 'm']);
0203 ANSI_REVERSE    : [VOLATILE] STRING4 := ('(27)', [' ', '7', 'm']);
0204
0205 VID_STRING4      : STRING4;
0206 NULL_STRING4     : STRING4 := ('(0)', '(0)', '(0)', '(0)');
0207 EMPTY_STRING    : STRING2 := ('', '');
0208 SHIFT           : [VOLATILE] STRING4 := ('(9)', '(0)', '(0)', '(0)');
0209 CRLF_SHIFT       : [VOLATILE] STRING6 := ('(13)', '(10)', '(9)', '(0)', '(0)', '(0)');
0210 LOW_SHIFT       : STRING3;
0211
0212 NULL_CHAR        : CHAR := ('(0)');
0213
0214 { +
0215 This is the terminal and screen database.
0216 - }
0217 TAB              : CHAR := ('(9)');
0218 ESCAPE           : [VOLATILE] CHAR := ('(27)');
0219 APOSTROPHE       : CHAR := ('(39)');
0220 CONTROL_W        : CHAR := ('(23)');
0221 CONTROL_Z        : [VOLATILE] CHAR := ('(26)');
0222 QUESTION_MARK    : [VOLATILE] CHAR := ('(63)');
0223 ERR_CHAR         : [VOLATILE] CHAR;
0224 CONTROL_G        : CHAR := ('(7)');
0225 CRLF            : [VOLATILE] STRING2 := ('(13)', '(10)');
0226 TERMINAL_TYPE    : BYTE;
0227 TERMINAL_SPEED   : INTEGER;
0228 ANSI_CRT         : BOOLEAN := FALSE;
0229 REGIS            : [VOLATILE] BOOLEAN := FALSE;
0230 DEC_CRT          : BOOLEAN := FALSE;
0231 DEV_TYPE         : INTEGER;
0232 VIDEO_TERMINAL   : BOOLEAN := FALSE;
0233 VID_TERM         : INTEGER;
0234 SCREEN_FLAGS     : [VOLATILE] SCR1$TYPE;
0235 OUT_LINE         : [VOLATILE] TEMP_VARYING;
0236 ONE             : [VOLATILE] INTEGER := 1;
0237 CHFFLAGS        : [VOLATILE] INTEGER := 0;
0238 FLAGS           : [VOLATILE] FDL2$TYPE;
0239 TEMP_FDL3$TYPE  : [VOLATILE] FDL3$TYPE;
0240 LINE_WIDTH      : INTEGER;
0241 LINES_PER_PAGE  : [VOLATILE] INTEGER;
0242 DEST_IS_TERMINAL : [VOLATILE] BOOLEAN;
0243 LINES_SHOWN     : [VOLATILE] INTEGER;
0244 MINIMUM_TERM_WIDTH : INTEGER := 80;
0245 MINIMUM_VIDEO_PAGE : INTEGER := 24;
0246 SCROLLING_SET   : [VOLATILE] BOOLEAN := FALSE;
0247 FULL_PROMPT     : BOOLEAN := TRUE;
0248 TEMP_FULL_PROMPT : [VOLATILE] BOOLEAN := FALSE;
0249 ORIG_TIME       : REAL;
0250 QUAD_TIME       : QUADWORD;
0251
0252 DEFAULT_PRIMARY : PRIMARY_TYPE := FILES;
0253 DEFAULT_PRINUM  : INTEGER := 0;
```

```
0256 COL ONE           : INTEGER := 1;
0257 LINE ONE         : [VOLATILE] INTEGER := 1;
0258 LOWER_LINE       : INTEGER := 17;
0259 PROMPT_LINE       : INTEGER := 23;
0260
0261 PARAM_BLOCK        : [VOLATILE]TPASTYPE;
0262
0263 SEC_ATTR           : STRING22 := ' Secondary Attributes ';
0264 EDFRLP_STRING       : STRING6 := 'EDFHLP';
0265 IDENT_STRING        : STRING40 :=
0266                                     VAX-11 FDL Editor';
0267 IDENT_STRING_LENGTH : INTEGER := 40;
0268 QUES_HINT           : STRING31 := '(Type "?" for list of Keywords)';
0269 EDF_HEADER          : STRING19 := ' VAX-11 FDL Editor ';
0270 CONTINUE_TEXT       : STRING45 :=
0271                                     ' Press RETURN to continue (^Z for Main Menu) ';
0272 ISTATUS             : [VOLATILE] INTEGER;
0273 FAB_DUMMY           : FAB_PTR;
0274 RAB_DUMMY           : FAB_PTR;
0275
0276 FDL_BLOCK           : [VOLATILE] ^FDL_TYPE;
0277 FDL$AL_BLOCK        : [EXTERNAL,VOLATILE] INTEGER;
0278
0279 EDF$GL_SECNUM        : [EXTERNAL,VOLATILE] LONG;
0280 EDF$GL_PROT_MASK     : [EXTERNAL] CTRL_ARRAY;
0281 EDF$GL_FID1          : [EXTERNAL,VOLATILE] LONG;
0282 EDF$GL_FID2          : [EXTERNAL,VOLATILE] LONG;
0283 EDF$GL_FID3          : [EXTERNAL,VOLATILE] LONG;
0284 EDF$GL_OWNER_UIC     : [EXTERNAL,VOLATILE] LONG;
0285 EDF$GL_SPARET        : [EXTERNAL,VOLATILE] LONG;
0286 EDF$AB_STRING        : [EXTERNAL,VOLATILE] DESCRIPTOR;
0287 EDF$AB_COMMENT       : [EXTERNAL,VOLATILE] DESCRIPTOR;
0288 EDF$AB_UIC_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0289 EDF$AB_UIC_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0290 EDF$AB_FID_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0291 EDF$AB_FID_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0292 EDF$AB_PRIMARY_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0293 EDF$AB_PRIMARY_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0294 EDF$AB_ACCESS_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0295 EDF$AB_ACCESS_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0296 EDF$AB_ACL_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0297 EDF$AB_ACL_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0298 EDF$AB_AREA_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0299 EDF$AB_AREA_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0300 EDF$AB_CONNECT_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0301 EDF$AB_CONNECT_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0302 EDF$AB_DATE_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0303 EDF$AB_DATE_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0304 EDF$AB_FILE_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0305 EDF$AB_FILE_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0306 EDF$AB_JOURNAL_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0307 EDF$AB_JOURNAL_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0308 EDF$AB_KEY_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0309 EDF$AB_KEY_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0310 EDF$AB_RECORD_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0311 EDF$AB_RECORD_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0312
```



```
0313 EDF$AB_SHARING_TABLE_KEY      : [EXTERNAL,VOLATILE] LONG;  
0314 EDF$AB_SHARING_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0315 EDF$AB_SYSTEM_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0316 EDF$AB_SYSTEM_TABLE_STA     : [EXTERNAL,VOLATILE] LONG;  
0317 EDF$AB_POSIT_TABLE_KEY      : [EXTERNAL,VOLATILE] LONG;  
0318 EDF$AB_POSIT_TABLE_STA      : [EXTERNAL,VOLATILE] LONG;  
0319 EDF$AB_PROT_TABLE_KEY       : [EXTERNAL,VOLATILE] LONG;  
0320 EDF$AB_PROT_TABLE_STA       : [EXTERNAL,VOLATILE] LONG;  
0321 EDF$AB_ORG_TABLE_KEY        : [EXTERNAL,VOLATILE] LONG;  
0322 EDF$AB_ORG_TABLE_STA        : [EXTERNAL,VOLATILE] LONG;  
0323 EDF$AB_SOURCE_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0324 EDF$AB_SOURCE_TABLE_STA     : [EXTERNAL,VOLATILE] LONG;  
0325 EDF$AB_RU_TABLE_KEY        : [EXTERNAL,VOLATILE] LONG;  
0326 EDF$AB_RU_TABLE_STA        : [EXTERNAL,VOLATILE] LONG;  
0327 EDF$AB_CARR_TABLE_KEY      : [EXTERNAL,VOLATILE] LONG;  
0328 EDF$AB_CARR_TABLE_STA      : [EXTERNAL,VOLATILE] LONG;  
0329 EDF$AB_FORMAT_TABLE_KEY    : [EXTERNAL,VOLATILE] LONG;  
0330 EDF$AB_FORMAT_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0331 EDF$AB_TYPE_TABLE_KEY       : [EXTERNAL,VOLATILE] LONG;  
0332 EDF$AB_TYPE_TABLE_STA      : [EXTERNAL,VOLATILE] LONG;  
0333 EDF$AB_LOAD_METHOD_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0334 EDF$AB_LOAD_METHOD_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0335 EDF$AB_WEIGHT_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0336 EDF$AB_WEIGHT_TABLE_STA     : [EXTERNAL,VOLATILE] LONG;  
0337 EDF$AB_SURFACE_OPTION_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0338 EDF$AB_SURFACE_OPTION_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0339 EDF$AB_CURRENT_FUNC_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0340 EDF$AB_CURRENT_FUNC_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0341 EDF$AB_DESIGN_CYCLE_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0342 EDF$AB_DESIGN_CYCLE_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0343 EDF$AB_SCRIPT_OPTION_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0344 EDF$AB_SCRIPT_OPTION_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0345 EDF$AB_YES_NO_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0346 EDF$AB_YES_NO_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0347 EDF$AB_SET_FUNCTION_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0348 EDF$AB_SET_FUNCTION_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0349 EDF$AB_GRANULARITY_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0350 EDF$AB_GRANULARITY_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0351 EDF$AB_PROMPTING_TABLE_KEY  : [EXTERNAL,VOLATILE] LONG;  
0352 EDF$AB_PROMPTING_TABLE_STA  : [EXTERNAL,VOLATILE] LONG;  
0353 EDF$AB_RESPONSES_TABLE_KEY  : [EXTERNAL,VOLATILE] LONG;  
0354 EDF$AB_RESPONSES_TABLE_STA  : [EXTERNAL,VOLATILE] LONG;
```

```
0355 ( +  
0356 The following are the pointers to the Definition Linked List.  
0357 - )
```

```
0359 DEF_CURRENT      : [VOLATILE] ^LINE_OBJECT := NIL;  
0360 DEF_SCRATCH      : [VOLATILE] ^LINE_OBJECT := NIL;  
0361 DEF_HEAD         : [VOLATILE] ^LINE_OBJECT := NIL;  
0362 DEF_TAIL         : [VOLATILE] ^LINE_OBJECT := NIL;  
0363 DEF_SUCC         : [VOLATILE] ^LINE_OBJECT := NIL;  
0364 DEF_PRED        : [VOLATILE] ^LINE_OBJECT := NIL;
```

```
0365 ( +  
0366 These point to the analysis linked list.  
0367 - )
```

```
0369 DEF_ANL_HEAD      : ^LINE_OBJECT := NIL;
```

```
0370 DEF_ANL_TAIL      : ^LINE_OBJECT := NIL;
0371 DEF_SAVE_HEAD    : ^LINE_OBJECT := NIL;
0372 DEF_SAVE_TAIL     : ^LINE_OBJECT := NIL;
0373
0374 POINTING_AT_DEFINITION : BOOLEAN := TRUE;
0375
0376 FILE_CREATED       : [VOLATILE] BOOLEAN := FALSE;
0377
0378 ( +
0379 These are used for input string processing.
0380 - )
0381 INPUT_DESC          : [VOLATILE] DESCRIPTOR;
0382 INPUT_STRING        : [VOLATILE] STRING255;
0383 INPUT_VALUE         : INTEGER;
0384 INPUT_NUMBER        : INTEGER;
0385
0386 QUAD_DESC           : [VOLATILE] RECORD
0387     CASE QWHICH : QD SWITCH OF
0388         QWORD      : (TWO LONG : PACKED RECORD
0389             L1, L2 : LONG
0390             END);
0391         DWORD      : (DSC : DESCRIPTOR)
0392     END;
0393
0394 ( +
0395 ACTIVE CALCULATION DATABASE.
0396 - )
0397 LINKED               : BOOLEAN;
0398 ACTIVE_AREA          : INTEGER;
0399 ACTIVE_PRIMARY       : PRIMARY_TYPE;
0400 VARIABLE_RECORDS     : BOOLEAN;
0401 CUR_MAX_REC          : INTEGER;
0402 BYTES_PER_BUCKET     : INTEGER;
0403 BUCKET_DEFAULT       : INTEGER;
0404 PRIMARY_INDEX_BUCKETS : INTEGER;
0405 INIT_PRIMARY_BUCKETS : ARRAY [0..31] OF INTEGER;
0406 ADDED_PRIMARY_BUCKETS : ARRAY [0..31] OF INTEGER;
0407 INIT_NUMBER_BUCKETS  : ARRAY [0..31] OF INTEGER;
0408 ADDED_NUMBER_BUCKETS : ARRAY [0..31] OF INTEGER;
0409 RECS_PER_BUCKET      : ARRAY [0..31] OF INTEGER;
0410 DEEPEST              : INTEGER;
0411 FIRST_PLOT           : BOOLEAN;
0412 OPTIMIZING           : BOOLEAN;
0413 VISIBLE_QUESTION     : BOOLEAN;
0414 WAIT_HELP            : BOOLEAN;
```

```
0416 ( +
0417 QTAB is the table that drives the Q+A routine - QUERY
0418 The xDATA arrays hold the main EDF database.
0419 - )
0420 QTAB      : ARRAY [EDFSK_QTABSTART..EDFSK_QTABEND] OF QTAB_ENTRY;
0421 QTAB_OFFSET : INTEGER;
0422
0423 ( +
0424 String descriptor elements
0425 - )
0426 SDATA      : ARRAY [EDFSK_SDATASTART..EDFSK_SDATAEND] OF DESCRIPTOR;
0427
0428 ( +
0429 Real elements
0430 - )
0431 RDATA      : ARRAY [EDFSK_RDATASTART..EDFSK_RDATAEND] OF REAL;
0432
0433 ( +
0434 Boolean elements
0435 - )
0436 BDATA      : ARRAY [EDFSK_BDATASTART..EDFSK_BDATAEND] OF BOOLEAN;
0437
0438 ( +
0439 Integer elements
0440 - )
0441 IDATA      : ARRAY [EDFSK_IDATASTART..EDFSK_IDATAEND] OF INTEGER;
0442
0443 ( +
0444 Valid (boolean) elements
0445 - )
0446 VDATA      : ARRAY [EDFSK_VDATASTART..EDFSK_VDATAEND] OF BOOLEAN;
0447
0448 ( +
0449 Misc. scratch variables used during the Q+A.
0450 - )
0451 TEMP_STRING255      : STRING255;
0452 TEMP_DESCRIPTOR     : [VOLATILE] DESCRIPTOR;
0453 QUERY_FLAG          : BOOLEAN;
0454 LOW_KEY              : INTEGER := 0;
0455 HIGH_KEY             : INTEGER := 0;
0456 LOW_AREA             : INTEGER := 0;
0457 HIGH_AREA            : INTEGER := 0;
0458 FOUND_AREA           : BOOLEAN := FALSE;
0459 FOUND_KEY             : BOOLEAN := FALSE;
0460 FOUND_O              : BOOLEAN := FALSE;
0461 MAX_KEY_SIZE         : INTEGER;
0462 MIN_KEY_SIZE         : INTEGER;
0463 SEGMENT_WANTED       : ARRAY [0..7] OF BOOLEAN :=
0464 (FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE);
0465 SEGMENT_POSITION     : ARRAY [0..7] OF INTEGER;
0466 SEGMENT_LENGTH       : ARRAY [0..7] OF INTEGER;
0467 SEGMENT_NUMBER       : INTEGER;
0468 BUCKET_OVERHEAD      : INTEGER;
0469 MIN_BUCKET           : INTEGER;
0470 ENTRY_SIZE           : INTEGER;
0471 LOWMAX               : INTEGER;
0472 EXTRA                : INTEGER;
```



```
0473 CUR_MAX_FIXED          : INTEGER;
0474 MAX_STRING_ANSWER_LENGTH : INTEGER;
0475 OLD_COUNT              : INTEGER;
0476 GLOBAL_SET            : BOOLEAN;
0477 NUMBER_KEYS_SET       : BOOLEAN := FALSE;
0478 ISAM_ORG              : BOOLEAN;
0479 MAX_KEY_POSITION      : INTEGER;
0480 TEMP_REAL             : REAL;
0481 TEMP_STATUS          : INTEGER;
0482 TEMP_INT2            : INTEGER;
0483 DEF                  : INTEGER;
0484 NULL_STRING          : [VOLATILE]DESCRIPTOR;
0485 TEST                 : LINE_OBJECT;
0486 FULL_CHOICE          : BOOLEAN;
0487
0488 ( *
0489 This is the array of secondary value types.
0490 - )
0491 SEC_TYPE              : [VOLATILE] PACKED ARRAY [SECONDARY_TYPE] OF
0492                        PACKED RECORD
0493                        STR      : BOOLEAN;
0494                        NUM      : BOOLEAN;
0495                        QUAL     : BOOLEAN;
0496                        SW       : BOOLEAN;
0497                        END;
0498
0499 ( *
0500 This is the template for line_objects generated by MAKE_SCRATCH.
0501 - )
0502 LINE_OBJECT_TEMPLATE  : [VOLATILE] LINE_OBJECT;
```

0504 VALUE
0505 %INCLUDE 'SRCS:EDFVALUE.PAS'
0506 { **
0507

0508 FILE: SRCS:EDFVALUE.PAS - Pascal include file to define
0509 initial values of selected top-level variables.
0510

0511 *****
0512 *
0513 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0514 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0515 * ALL RIGHTS RESERVED.
0516 *
0517 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0518 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0519 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0520 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0521 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0522 * TRANSFERRED.
0523 *
0524 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0525 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0526 * CORPORATION.
0527 *
0528 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0529 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0530 *
0531 *****

0532
0533
0534
0535
0536
0537
0538 FACILITY: VAX/VMS EDF (EDIT/FDL) UTILITY
0539

0540 ABSTRACT: This facility is used to create, modify, and optimize
0541 FDL specification files.
0542

0543 ENVIRONMENT: NATIVE/USER MODE
0544

0545 AUTHOR: Ken F. Henderson Jr.
0546

0547 CREATION DATE: 27-Mar-1981
0548

0549 MODIFIED BY:
0550

0551 V03-009 KFH0009 Ken Henderson 10 Sep 1983
0552 Support named UICs.
0553

0554 V03-008 KFH0008 Ken Henderson 9 Aug 1983
0555 Fix max value of CLUSTER_SIZE.
0556 Fix default of QTAB[TEST_PRIMARY].
0557

0558 V03-007 KFH0007 Ken Henderson 30 Jul 1983
0559 Fix SEC_TYPE table for audit_trail.
0560 Add DEFERRED_WRITE.

V03-006	KFH0006	Ken Henderson	26 Apr 1983
	Fix various defaults in QTAB. Transferred some initializations to the EDFVAR declarations.		
V03-005	KFH0005	Ken Henderson	14 Apr 1983
	Changed max bucket size to 63 from 65. Added ANALYSIS, OUTPUT, RESPONSES, PROMPTING, SET FUNCTION, GRANULARITY. Added support for SEGMENTED keys.		
V03-004	KFH0004	Ken Henderson	7 Mar 1983
	Changed max bucket size to 65 from 127.		
V03-003	KFH0003	Ken Henderson	11 Sept 1982
	Added initialization of VDATA and BDATA.		
V03-002	KFH0002	Ken Henderson	9 Sept 1982
	Added initialization of QTAB.		
V03-001	KFH0001	Ken Henderson	23-Mar-1982
	Took out reference to EDITFDL_STRING		

--)


```
0587 { +  
0588 Initialize the Boolean-array to all false.  
0589 - }  
0590 BDATA := (  
0591 FALSE,  
0592 FALSE,  
0593 FALSE,  
0594 FALSE,  
0595 FALSE,  
0596 FALSE,  
0597 FALSE,  
0598 FALSE,  
0599 FALSE,  
0600 FALSE,  
0601 FALSE,  
0602 FALSE,  
0603 FALSE,  
0604 FALSE,  
0605 FALSE,  
0606 FALSE,  
0607 FALSE,  
0608 FALSE,  
0609 FALSE,  
0610 FALSE,  
0611 FALSE,  
0612 FALSE,  
0613 FALSE,  
0614 FALSE,  
0615 FALSE,  
0616 );  
0617  
0618 { +  
0619 Initialize the String-array to all null-string.  
0620 - }  
0621 SDATA := (  
0622  
0623 (0,DSCBK_DTYPE_T,DSCBK_CLASS_D,NIL),  
0624 (0,DSCBK_DTYPE_T,DSCBK_CLASS_D,NIL),  
0625 (0,DSCBK_DTYPE_T,DSCBK_CLASS_D,NIL),  
0626 (0,DSCBK_DTYPE_T,DSCBK_CLASS_D,NIL),  
0627 (0,DSCBK_DTYPE_T,DSCBK_CLASS_D,NIL),  
0628 (0,DSCBK_DTYPE_T,DSCBK_CLASS_D,NIL),  
0629  
0630 );  
0631  
0632 { +  
0633 Initialize the Valid-array to all false.  
0634 - }  
0635 VDATA := (  
0636 FALSE,  
0637 FALSE,  
0638 FALSE,  
0639 FALSE,  
0640 FALSE,  
0641 FALSE,  
0642 FALSE,  
0643 FALSE,
```

EDF VAR
VOL-000

Source Listing

16-Sep-1984 00:42:36
15-Sep-1984 22:43:40

VAX-11 Pascal V2.4-277 Page 14
_S2558DUA28:[EDF.SRC]EDFVALUE.PAS:1 (2)

0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700

[illegible][illegible]

```
0701 { +
0702 Initialize the sequencing array.
0703 - }
0704 PRI_SEQ := (
0705     15,      { DUMMY_PRIMARYS }
0706     8,       { ACCESS, }
0707     4,       { ACL, }
0708     13,      { ANALYSIS_OF_AREA, }
0709     14,      { ANALYSIS_OF_KEY, }
0710     11,      { AREA, }
0711     10,      { CONNECT, }
0712     6,       { DATE, }
0713     3,       { FILES, }
0714     1,       { IDENT, }
0715     6,       { JOURNAL, }
0716     12,      { KEY, }
0717     7,       { RECORDS, }
0718     9,       { SHARING, }
0719     2,       { SYSTEM, }
0720     0,       { TITLE }
0721 );
0722
0723 { +
0724 Initialize the 'width' arrays - that indicate how long a particular
0725 keyword should be printed.
0726 - }
0727 PRIMARY_WIDTH := (
0728     0,      { DUMMY_PRIMARYS }
0729     6,      { ACCESS, }
0730     3,      { ACL, }
0731     16,     { ANALYSIS_OF_AREA, }
0732     15,     { ANALYSIS_OF_KEY, }
0733     6,      { AREA, }
0734     7,      { CONNECT, }
0735     6,      { DATE, }
0736     6,      { FILES, }
0737     5,      { IDENT, }
0738     7,      { JOURNAL, }
0739     3,      { KEY, }
0740     9,      { RECORDS, }
0741     7,      { SHARING, }
0742     8,      { SYSTEM, }
0743     3,      { TITLE }
0744 );
0745
0746 SECONDARY_WIDTH := (
0747
0748 { RESERVE 0 } 0,      { DUMMY_SECONDARYS, }
0749
0750 { ACCESS PRIMARY }
0751
0752     8,      { BLOCK_IO$ }
0753     6,      { DELETES }
0754     3,      { GET$ }
0755     3,      { PUT$ }
```



```
0758 1 { RECORD IOS }
0759 8 { TRUNCATES }
0760 6 { UPDATES }
0761
0762 { ACL PRIMARY }
0763
0764 5 { ENTRY }
0765
0766 { ANALYSIS_OF_AREA PRIMARY }
0767 15 { RECLAIMED_SPACE }
0768
0769 { ANALYSIS_OF_KEY PRIMARY }
0770
0771 9 { DATA_FILLS, }
0772 20 { DATA_KEY_COMPRESSION, }
0773 23 { DATA_RECORD_COMPRESSION, }
0774 17 { DATA_RECORD_COUNT, }
0775 19 { DATA_SPACE_OCCUPIED, }
0776 9 { DELETIONS, }
0777 5 { DEPTH, }
0778 16 { DUPLICATES_PER_SIDR, }
0779 17 { INDEX_COMPRESSION, }
0780 10 { INDEX_FILLS, }
0781 20 { INDEX_SPACE_OCCUPIED, }
0782 19 { LEVELT_RECORD_COUNT, }
0783 16 { MEAN_DATA_LENGTH, }
0784 17 { MEAN_INDEX_LENGTH, }
0785 15 { RANDOM_ACCESSES, }
0786 14 { RANDOM_INSERTS, }
0787 19 { SEQUENTIAL_ACCESSES, }
0788
0789 { AREA PRIMARY }
0790
0791 10 { ALLOCATIONS, }
0792 19 { BEST_TRY_CONTIGUOUS, }
0793 11 { BUCKET_SIZES, }
0794 10 { CONTIGUOUS, }
0795 17 { EXACT_POSITIONINGS, }
0796 9 { EXTENSIONS, }
0797 8 { POSITIONS, }
0798 6 { VOLUMES, }
0799
0800 { CONNECT PRIMARY }
0801
0802 12 { ASYNCHRONOUS }
0803 8 { BLOCK_IO }
0804 11 { BUCKET_CODE }
0805 7 { CONTEXT }
0806 11 { END_OF_FILE }
0807 12 { FILE_BUCKETS }
0808 11 { FAST_DELETE }
0809 16 { KEY_OF_REFERENCE }
0810 17 { KEY_GREATER_EQUAL }
0811 16 { KEY_GREATER_THAN }
0812 9 { KEY_LIMIT }
0813 11 { LOCATE_MODE }
0814 12 { LOCK_OR_READ }
```

```
0815 I 13. { LOCK ON WRITE }
0816 I 16. { MANUAL UNLOCKING }
0817 I 16. { MULTIBLOCK_COUNT }
0818 I 17. { MULTIBUFFER_COUNT }
0819 I 6. { NOLOCK }
0820 I 18. { NONEXISTENT_RECORD }
0821 I 10. { READ_AHEAD }
0822 I 15. { READ-REGARDLESS }
0823 I 14. { TIMEOUT_ENABLE }
0824 I 14. { TIMEOUT_PERIOD }
0825 I 15. { TRUNCATE ON PUT }
0826 I 19. { TT_CANCEL_CONTROL_0 }
0827 I 15. { TT_UPCASE_INPUT }
0828 I 9. { TT_PROMPT }
0829 I 19. { TT_PURGE_TYPE_AHEAD }
0830 I 14. { TT_READ_NOECHO }
0831 I 16. { TT_READ_NOFILTER }
0832 I 9. { UPDATE IF }
0833 I 15. { WAIT FOR RECORD }
0834 I 12. { WRITE_BEHIND }
0835
0836 I { DATE PRIMARY }
0837 I
0838 I 6. { BACKUPS, }
0839 I 8. { CREATIONS, }
0840 I 10. { EXPIRATIONS, }
0841 I 8. { REVISIONS, }
0842
0843 I { FILE PRIMARY }
0844 I
0845 I 10. { ALLOCATION, }
0846 I 19. { BEST TRY CONTIGUOUS, }
0847 I 11. { BUCKET SIZE, }
0848 I 12. { CLUSTER SIZE, }
0849 I 7. { CONTEXTS }
0850 I 10. { CONTIGUOUS, }
0851 I 9. { CREATE IF }
0852 I 12. { DEFAULT NAME, }
0853 I 14. { DEFERRED WRITE, }
0854 I 15. { DELETE ON CLOSE, }
0855 I 15. { DIRECTORY_ENTRY, }
0856 I 15. { ERASE ON DELETE, }
0857 I 9. { EXTENSION, }
0858 I 19. { GLOBAL BUFFER_COUNT, }
0859 I 13. { MT_BLOCK_SIZE, }
0860 I 19. { MT_CURRENT_POSITION, }
0861 I 10. { MT_NOT_EOF }
0862 I 13. { MT_PROTECTION, }
0863 I 14. { MT_OPEN_REWIND, }
0864 I 15. { MT_CLOSE_REWIND }
0865 I 17. { MAX_RECORD_NUMBER, }
0866 I 16. { MAXIMIZE_VERSION, }
0867 I 4. { NAME, }
0868 I 8. { NOBACKUP, }
0869 I 19. { NON FILE STRUCTURED }
0870 I 17. { OUTPUT FILE PARSE }
0871 I 12. { ORGANIZATION, }
```

```
0872 1 { OWNER, }
0873 1 { PRINT_ON_CLOSE, }
0874 1 { PROTECTION, }
0875 1 { READ_CHECK, }
0876 1 { REVISION, }
0877 1 { SEQUENTIAL_ONLY, }
0878 1 { SUBMIT_ON_CLOSE, }
0879 1 { SUPERSEDE, }
0880 1 { TEMPORARY, }
0881 1 { TRUNCATE_ON_CLOSE, }
0882 1 { USER_FILE_OPEN, }
0883 1 { WINDOW_SIZE, }
0884 1 { WRITE_CHECK, }
0885
0886 { JOURNALING PRIMARY }
0887
0888 11 { AFTER_IMAGE, }
0889 10 { AFTER_NAME, }
0890 11 { AUDIT-TRAIL, }
0891 10 { AUDIT_NAME, }
0892 12 { BEFORE_IMAGE, }
0893 11 { BEFORE_NAME, }
0894 13 { RECOVERY_UNIT, }
0895
0896 { KEY PRIMARY }
0897
0898 7 { CHANGES, }
0899 9 { DATA_AREA, }
0900 9 { DATA_FILL, }
0901 20 { DATA_KEY_COMPRESSION, }
0902 23 { DATA_RECORD_COMPRESSION, }
0903 10 { DUPLICATES, }
0904 10 { INDEX_AREA, }
0905 17 { INDEX_COMPRESSION, }
0906 10 { INDEX_FILL, }
0907 17 { LEVELT_INDEX_AREA, }
0908 4 { NAMES, }
0909 8 { NULL_KEY, }
0910 10 { NULL_VALUE, }
0911 6 { PROLOG(UE) - 1ST 6 CHARS ONLY }
0912 0 { SEG_LENGTH, }
0913 0 { SEG_POSITION, }
0914 0 { SEG_TYPE, }
0915
0916 { RECORD PRIMARY }
0917
0918 10 { BLOCK_SPAN, }
0919 16 { CARRIAGE_CONTROL, }
0920 18 { CONTROL_FIELD_SIZE, }
0921 6 { FORMAT, }
0922 4 { SIZE, }
0923
0924 { SHARING PRIMARY }
0925
0926 6 { DELETE }
0927 3 { GET }
0928 11 { MULTISTREAM }
```



```
0929 I      8,      { PROHIBIT }
0930 I      3,      { PUT }
0931 I      6,      { UPDATE }
0932 I     14,      { USER_INTERLOCK }
0933 I
0934 I { SYSTEM PRIMARY }
0935 I
0936 I      6,      { DEVICE, }
0937 I      6,      { SOURCE, }
0938 I      6,      { TARGET, }
0939 I
0940 I      );
0941 I
0942 I { +
0943 I   These are the maximum values of number-valued secondaries.
0944 I - }
0945 I
0946 I SECONDARY_MAX := (
0947 I { RESERVE 0 }      0,      { DUMMY_SECONDARY$, }
0948 I { ACCESS PRIMARY }
0949 I
0950 I
0951 I      0,      { BLOCK_IO$, }
0952 I      0,      { DELETES, }
0953 I      0,      { GET$, }
0954 I      0,      { PUT$, }
0955 I      0,      { RECORD_IO$, }
0956 I      0,      { TRUNCATES, }
0957 I      0,      { UPDATES, }
0958 I
0959 I { ACL PRIMARY }
0960 I
0961 I      0,      { ENTRY }
0962 I
0963 I { ANALYSIS_OF_AREA PRIMARY }
0964 I      0,      { RECLAIMED_SPACE }
0965 I
0966 I { ANALYSIS_OF_KEY PRIMARY }
0967 I
0968 I      0,      { DATA_FILLS, }
0969 I      0,      { DATA_KEY_COMPRESSION, }
0970 I      0,      { DATA_RECORD_COMPRESSION, }
0971 I      0,      { DATA_RECORD_COUNT, }
0972 I      0,      { DATA_SPACE_OCCUPIED, }
0973 I      0,      { DELETIONS, }
0974 I      0,      { DEPTH, }
0975 I      0,      { DUPLICATES_PER_SIDR, }
0976 I      0,      { INDEX_COMPRESSION, }
0977 I      0,      { INDEX_FILLS, }
0978 I      0,      { INDEX_SPACE_OCCUPIED, }
0979 I      0,      { LEVELT_RECORD_COUNT, }
0980 I      0,      { MEAN_DATA_LENGTH, }
0981 I      0,      { MEAN_INDEX_LENGTH, }
0982 I      0,      { RANDOM_ACCESSES, }
0983 I      0,      { RANDOM_INSERTS, }
0984 I      0,      { SEQUENTIAL_ACCESSES, }
0985 I
```

```
0986 |
0987 | ( AREA PRIMARY )
0988 |
0989 | EDFSC_1GIGA, { ALLOCATIONS, }
0990 | 0, { BEST_TRY_CONTIGUOUS, }
0991 | BKTSC_MAXBKTSIZ, { BUCKET_SIZES, }
0992 | 0, { CONTIGUOUS, }
0993 | 0, { EXACT_POSITIONINGS, }
0994 | EDFSC_1GIGA, { EXTENSIONS, }
0995 | 16777215, { POSITIONS, }
0996 | 65535, { VOLUMES, }
0997 |
0998 | ( CONNECT PRIMARY )
0999 |
1000 | 0, { ASYNCHRONOUS }
1001 | 0, { BLOCK_IO }
1002 | EDFSC_1GIGA, { BUCKET_CODE }
1003 | EDFSC_1GIGA, { CONTEXT }
1004 | 0, { END_OF_FILE }
1005 | 0, { FILE_BUCKETS }
1006 | 0, { FAST_DELETE }
1007 | 255, { KEY_OF_REFERENCE }
1008 | 0, { KEY_GREATER_EQUAL }
1009 | 0, { KEY_GREATER_THAN }
1010 | 0, { KEY_LIMIT }
1011 | 0, { LOCATE_MODE }
1012 | 0, { LOCK_ON_READ }
1013 | 0, { LOCK_ON_WRITE }
1014 | 0, { MANUAL_UNLOCKING }
1015 | 255, { MULTIBLOCK_COUNT }
1016 | 255, { MULTIBUFFER_COUNT }
1017 | 0, { NOLOCK }
1018 | 0, { NONEXISTENT_RECORD }
1019 | 0, { READ_AHEAD }
1020 | 0, { READ_REGARDLESS }
1021 | 0, { TIMEOUT_ENABLE }
1022 | 255, { TIMEOUT_PERIOD }
1023 | 0, { TRUNCATE_ON_PUT }
1024 | 0, { TT_CANCEL_CONTROL_0 }
1025 | 0, { TT_UPCASE_INPUT }
1026 | 0, { TT_PROMPT }
1027 | 0, { TT_PURGE_TYPE_AHEAD }
1028 | 0, { TT_READ_NOECHO }
1029 | 0, { TT_READ_NOFILTER }
1030 | 0, { UPDATE_IF }
1031 | 0, { WAIT_FOR_RECORD }
1032 | 0, { WRITE_BEHIND }
1033 |
1034 | ( DATE PRIMARY )
1035 |
1036 | 0, { BACKUPS, }
1037 | 0, { CREATIONS, }
1038 | 0, { EXPIRATIONS, }
1039 | 0, { REVISIONS, }
1040 |
1041 | ( FILE PRIMARY )
1042 |
```

```
1043 EDFSC_1GIGA, { ALLOCATION, }
1044 0, { BEST_TRY, CONTIGUOUS, }
1045 BKTSC_MAXBKTSIZ, { BUCKET_SIZE, }
1046 EDFSC_1GIGA, { CLUSTER_SIZE, }
1047 EDFSC_1GIGA, { CONTEXT }
1048 0, { CONTIGUOUS, }
1049 0, { CREATE_IF }
1050 0, { DEFAULT_NAME }
1051 0, { DEFERRED_WRITE }
1052 0, { DELETE_ON_CLOSE, }
1053 0, { DIRECTORY_ENTRY, }
1054 0, { ERASE_ON_DELETE, }
1055 EDFSC_1GIGA, { EXTENSION, }
1056 EDFSC_MAX_GBL_BUFS, { GLOBAL_BUFFER_COUNT, }
1057 65532, { MT_BLOCK_SIZE, }
1058 0, { MT_CURRENT_POSITION, }
1059 0, { MT_NOT_EOF }
1060 0, { MT_PROTECTION, }
1061 0, { MT_OPEN_REWIND, }
1062 0, { MT_CLOSE_REWIND }
1063 EDFSC_1GIGA, { MAX_RECORD_NUMBER, }
1064 0, { MAXIMIZE_VERSION, }
1065 0, { NAME, }
1066 0, { NOBACKUP, }
1067 0, { NON_FILE_STRUCTURED }
1068 0, { OUTPUT_FILE_PARSE }
1069 0, { ORGANIZATION, }
1070 0, { OWNER, }
1071 0, { PRINT_ON_CLOSE, }
1072 0, { PROTECTION, }
1073 0, { READ_CHECK, }
1074 65535, { REVISION, }
1075 0, { SEQUENTIAL_ONLY }
1076 0, { SUBMIT_ON_CLOSE, }
1077 0, { SUPERSEDE, }
1078 0, { TEMPORARY }
1079 0, { TRUNCATE_ON_CLOSE, }
1080 0, { USER_FILE_OPEN }
1081 EDFSC_1GIGA, { WINDOW_SIZE }
1082 0, { WRITE_CHECK, }
1083
1084 { JOURNALING PRIMARY }
1085
1086 0, { AFTER_IMAGE, }
1087 0, { AFTER_NAME }
1088 0, { AUDIT-TRAIL, }
1089 0, { AUDIT-NAME }
1090 0, { BEFORE_IMAGE, }
1091 0, { BEFORE-NAME }
1092 0, { RECOVERY_UNIT, }
1093
1094 { KEY PRIMARY }
1095
1096 0, { CHANGES, }
1097 254, { DATA_AREA, }
1098 100, { DATA_FILL, }
1099 99, { DATA_KEY_COMPRESSION, }
```

```
1100      99,      { DATA_RECORD_COMPRESSION, }
1101      0,      { DUPLICATES, }
1102      254,    { INDEX_AREA, }
1103      99,     { INDEX_COMPRESSION, }
1104      100,    { INDEX_FILL, }
1105      254,    { LEVELT_INDEX_AREA, }
1106      0,      { NAMES, }
1107      0,      { NULL_KEY, }
1108      255,    { NULL_VALUE, }
1109      0,      { PROLOGUE, }
1110      255,    { SEG_LENGTH, }
1111      16299,  { SEG_POSITION, }
1112      0,      { SEG_TYPE, }
1113
1114      { RECORD PRIMARY }
1115
1116      0,      { BLOCK_SPAN, }
1117      0,      { CARRIAGE_CONTROL, }
1118      255,    { CONTROL_FIELD_SIZE, }
1119      0,      { FORMAT, }
1120      EDFSK_MAXRECSIZ, { SIZE, }
1121
1122      { SHARING PRIMARY }
1123
1124      0,      { DELETE }
1125      0,      { GET }
1126      0,      { MULTISTREAM }
1127      0,      { PROHIBIT }
1128      0,      { PUT }
1129      0,      { UPDATE }
1130      0,      { USER_INTERLOCK }
1131
1132      { SYSTEM PRIMARY }
1133
1134      0,      { DEVICE, }
1135      0,      { SOURCE, }
1136      0,      { TARGET, }
1137
1138      );
1139
1140      { +
1141      These are the secondary value types.
1142      - }
1143
1144      SEC_TYPE := (
1145
1146      { +
1147      KEY:      STR, NUM, QUAL, SW
1148      - }
1149
1150      { RESERVE 0 }      (FALSE,FALSE,FALSE,FALSE),      { DUMMY_SECONDARY$, }
1151
1152      { ACCESS PRIMARY }
1153
1154      (FALSE,FALSE,FALSE,TRUE),      { BLOCK_IOS }
1155      (FALSE,FALSE,FALSE,TRUE),      { DELETES }
1156      (FALSE,FALSE,FALSE,TRUE),      { GETS }
```



```
1157      (FALSE,FALSE,FALSE,TRUE),      { PUT$ }
1158      (FALSE,FALSE,FALSE,TRUE),      { RECORD_IOS }
1159      (FALSE,FALSE,FALSE,TRUE),      { TRUNCATES }
1160      (FALSE,FALSE,FALSE,TRUE),      { UPDATES }
1161
1162      { ACL PRIMARY }
1163
1164      (TRUE,FALSE,FALSE,FALSE),      { ENTRY }
1165
1166      { ANALYSIS_OF_AREA PRIMARY }
1167
1168      (FALSE,FALSE,FALSE,FALSE),      { RECLAIMED_SPACE }
1169
1170      { ANALYSIS_OF_KEY PRIMARY }
1171
1172      (FALSE,FALSE,FALSE,FALSE),      { DATA_FILLS, }
1173      (FALSE,FALSE,FALSE,FALSE),      { DATA_KEY_COMPRESSION, }
1174      (FALSE,FALSE,FALSE,FALSE),      { DATA_RECORD_COMPRESSION, }
1175      (FALSE,FALSE,FALSE,FALSE),      { DATA_RECORD_COUNT, }
1176      (FALSE,FALSE,FALSE,FALSE),      { DATA_SPACE_OCCUPIED, }
1177      (FALSE,FALSE,FALSE,FALSE),      { DELETIONS, }
1178      (FALSE,FALSE,FALSE,FALSE),      { DEPTH, }
1179      (FALSE,FALSE,FALSE,FALSE),      { DUPLICATES_PER_SIDR, }
1180      (FALSE,FALSE,FALSE,FALSE),      { INDEX_COMPRESSION, }
1181      (FALSE,FALSE,FALSE,FALSE),      { INDEX_FILLS, }
1182      (FALSE,FALSE,FALSE,FALSE),      { INDEX_SPACE_OCCUPIED, }
1183      (FALSE,FALSE,FALSE,FALSE),      { LEVEL_RECORD_COUNT }
1184      (FALSE,FALSE,FALSE,FALSE),      { MEAN_DATA_LENGTH, }
1185      (FALSE,FALSE,FALSE,FALSE),      { MEAN_INDEX_LENGTH, }
1186      (FALSE,FALSE,FALSE,FALSE),      { RANDOM_ACCESSES, }
1187      (FALSE,FALSE,FALSE,FALSE),      { RANDOM_INSERTS, }
1188      (FALSE,FALSE,FALSE,FALSE),      { SEQUENTIAL_ACCESSES, }
1189
1190      { AREA PRIMARY }
1191
1192      (FALSE,TRUE,FALSE,FALSE),      { ALLOCATIONS, }
1193      (FALSE,FALSE,FALSE,TRUE),      { BEST_TRY_CONTIGUOUS, }
1194      (FALSE,TRUE,FALSE,FALSE),      { BUCKET_SIZES, }
1195      (FALSE,FALSE,FALSE,TRUE),      { CONTIGUOUS, }
1196      (FALSE,FALSE,FALSE,TRUE),      { EXACT_POSITIONINGS, }
1197      (FALSE,TRUE,FALSE,FALSE),      { EXTENSIONS, }
1198      (FALSE,FALSE,FALSE,FALSE),      { POSITIONS, }
1199      (FALSE,TRUE,FALSE,FALSE),      { VOLUMES, }
1200
1201      { +
1202      KEY:          STR,  NUM,  QUAL,  SW
1203      - }
1204
1205      { CONNECT PRIMARY }
1206
1207      (FALSE,FALSE,FALSE,TRUE),      { ASYNCHRONOUS }
1208      (FALSE,FALSE,FALSE,TRUE),      { BLOCK_IO }
1209      (FALSE,TRUE,FALSE,FALSE),      { BUCKET_CODE }
1210      (FALSE,TRUE,FALSE,FALSE),      { CONTEXT }
1211      (FALSE,FALSE,FALSE,TRUE),      { END_OF_FILE }
1212      (FALSE,FALSE,FALSE,TRUE),      { FILE_BUCKETS }
1213      (FALSE,FALSE,FALSE,TRUE),      { FAST_DELETE }
```

```
1214 (FALSE,TRUE,FALSE,FALSE),
1215 (FALSE,FALSE,FALSE,TRUE),
1216 (FALSE,FALSE,FALSE,TRUE),
1217 (FALSE,FALSE,FALSE,TRUE),
1218 (FALSE,FALSE,FALSE,TRUE),
1219 (FALSE,FALSE,FALSE,TRUE),
1220 (FALSE,FALSE,FALSE,TRUE),
1221 (FALSE,FALSE,FALSE,TRUE),
1222 (FALSE,TRUE,FALSE,FALSE),
1223 (FALSE,TRUE,FALSE,FALSE),
1224 (FALSE,FALSE,FALSE,TRUE),
1225 (FALSE,FALSE,FALSE,TRUE),
1226 (FALSE,FALSE,FALSE,TRUE),
1227 (FALSE,FALSE,FALSE,TRUE),
1228 (FALSE,FALSE,FALSE,TRUE),
1229 (FALSE,TRUE,FALSE,FALSE),
1230 (FALSE,FALSE,FALSE,TRUE),
1231 (FALSE,FALSE,FALSE,TRUE),
1232 (FALSE,FALSE,FALSE,TRUE),
1233 (FALSE,FALSE,FALSE,TRUE),
1234 (FALSE,FALSE,FALSE,TRUE),
1235 (FALSE,FALSE,FALSE,TRUE),
1236 (FALSE,FALSE,FALSE,TRUE),
1237 (FALSE,FALSE,FALSE,TRUE),
1238 (FALSE,FALSE,FALSE,TRUE),
1239 (FALSE,FALSE,FALSE,TRUE),
1240
1241 ( DATE PRIMARY )
1242
1243 (TRUE,FALSE,FALSE,FALSE),
1244 (TRUE,FALSE,FALSE,FALSE),
1245 (TRUE,FALSE,FALSE,FALSE),
1246 (TRUE,FALSE,FALSE,FALSE),
1247
1248 ( FILE PRIMARY )
1249
1250 (FALSE,TRUE,FALSE,FALSE),
1251 (FALSE,FALSE,FALSE,TRUE),
1252 (FALSE,TRUE,FALSE,FALSE),
1253 (FALSE,TRUE,FALSE,FALSE),
1254 (FALSE,TRUE,FALSE,FALSE),
1255 (FALSE,FALSE,FALSE,TRUE),
1256 (FALSE,FALSE,FALSE,TRUE),
1257 (TRUE,FALSE,FALSE,FALSE),
1258 (FALSE,FALSE,FALSE,TRUE),
1259 (FALSE,FALSE,FALSE,TRUE),
1260 (FALSE,FALSE,FALSE,TRUE),
1261 (FALSE,FALSE,FALSE,TRUE),
1262 (FALSE,TRUE,FALSE,FALSE),
1263 (FALSE,TRUE,FALSE,FALSE),
1264 (FALSE,TRUE,FALSE,FALSE),
1265 (FALSE,FALSE,FALSE,TRUE),
1266 (FALSE,FALSE,FALSE,TRUE),
1267 (FALSE,FALSE,FALSE,FALSE),
1268 (FALSE,FALSE,FALSE,TRUE),
1269 (FALSE,FALSE,FALSE,TRUE),
1270 (FALSE,TRUE,FALSE,FALSE),
```

```
{ KEY_OF_REFERENCE }
{ KEY_GREATER_EQUAL }
{ KEY_GREATER_THAN }
{ KEY_LIMIT }
{ LOCATE_MODE }
{ LOCK_ON_READ }
{ LOCK_ON_WRITE }
{ MANUAL_UNLOCKING }
{ MULTIBLOCK_COUNT }
{ MULTIBUFFER_COUNT }
{ NOLOCK }
{ NONEXISTENT_RECORD }
{ READ_AHEAD }
{ READ_REGARDLESS }
{ TIMEOUT_ENABLE }
{ TIMEOUT_PERIOD }
{ TRUNCATE_ON_PUT }
{ TT_CANCEL_CONTROL_0 }
{ TT_UPCASE_INPUT }
{ TT_PROMPT }
{ TT_PURGE_TYPE_AHEAD }
{ TT_READ_NOECHO }
{ TT_READ_NOFILTER }
{ UPDATE_IF }
{ WAIT_FOR_RECORD }
{ WRITE_BEHIND }
```

```
{ BACKUPS, }
{ CREATIONS, }
{ EXPIRATIONS, }
{ REVISIONS, }
```

```
{ ALLOCATION, }
{ BEST_TRY_CONTIGUOUS, }
{ BUCKET_SIZE, }
{ CLUSTER_SIZE, }
{ CONTEXTS }
{ CONTIGUOUS, }
{ CREATE_IF }
{ DEFAULT_NAME }
{ DEFERRED_WRITE }
{ DELETE_ON_CLOSE, }
{ DIRECTORY_ENTRY, }
{ ERASE_ON_DELETE, }
{ EXTENSION, }
{ GLOBAL_BUFFER_COUNT, }
{ MT_BLOCK_SIZE, }
{ MT_CURRENT_POSITION, }
{ MT_NOT_EOF }
{ MT_PROTECTION, }
{ MT_OPEN_REWIND, }
{ MT_CLOSE_REWIND }
{ MAX_RECORD_NUMBER, }
```

```
1271 (FALSE,FALSE,FALSE,TRUE),
1272 (TRUE,FALSE,FALSE,FALSE),
1273 (FALSE,FALSE,FALSE,TRUE),
1274 (FALSE,FALSE,FALSE,TRUE),
1275 (FALSE,FALSE,FALSE,TRUE),
1276 (FALSE,FALSE,TRUE,FALSE),
1277 (FALSE,FALSE,FALSE,FALSE),
1278 (FALSE,FALSE,FALSE,TRUE),
1279 (FALSE,FALSE,FALSE,FALSE),
1280 (FALSE,FALSE,FALSE,TRUE),
1281 (FALSE,TRUE,FALSE,FALSE),
1282 (FALSE,FALSE,FALSE,TRUE),
1283 (FALSE,FALSE,FALSE,TRUE),
1284 (FALSE,FALSE,FALSE,TRUE),
1285 (FALSE,FALSE,FALSE,TRUE),
1286 (FALSE,FALSE,FALSE,TRUE),
1287 (FALSE,FALSE,FALSE,TRUE),
1288 (FALSE,TRUE,FALSE,FALSE),
1289 (FALSE,FALSE,FALSE,TRUE),
1290
1291 ( +
1292 KEY: STR, NUM, QUAL, SW
1293 - )
1294
1295 { JOURNAL PRIMARY }
1296
1297 (FALSE,FALSE,FALSE,TRUE),
1298 (TRUE,FALSE,FALSE,FALSE),
1299 (FALSE,FALSE,FALSE,TRUE),
1300 (TRUE,FALSE,FALSE,FALSE),
1301 (FALSE,FALSE,FALSE,TRUE),
1302 (TRUE,FALSE,FALSE,FALSE),
1303 (FALSE,FALSE,TRUE,FALSE),
1304
1305 { KEY PRIMARY }
1306
1307 (FALSE,FALSE,FALSE,TRUE),
1308 (FALSE,TRUE,FALSE,FALSE),
1309 (FALSE,TRUE,FALSE,FALSE),
1310 (FALSE,FALSE,FALSE,TRUE),
1311 (FALSE,FALSE,FALSE,TRUE),
1312 (FALSE,FALSE,FALSE,TRUE),
1313 (FALSE,TRUE,FALSE,FALSE),
1314 (FALSE,FALSE,FALSE,TRUE),
1315 (FALSE,TRUE,FALSE,FALSE),
1316 (FALSE,TRUE,FALSE,FALSE),
1317 (TRUE,FALSE,FALSE,FALSE),
1318 (FALSE,FALSE,FALSE,TRUE),
1319 (FALSE,FALSE,FALSE,FALSE),
1320 (FALSE,TRUE,FALSE,FALSE),
1321 (FALSE,TRUE,FALSE,FALSE),
1322 (FALSE,TRUE,FALSE,FALSE),
1323 (FALSE,FALSE,TRUE,FALSE),
1324
1325 { RECORD PRIMARY }
1326
1327 (FALSE,FALSE,FALSE,TRUE),
```

```
{ MAXIMIZE_VERSION, }
{ NAME, }
{ NOBACKUP, }
{ NON_FILE_STRUCTURED, }
{ OUTPUT_FILE_PARSE, }
{ ORGANIZATION, }
{ OWNER, }
{ PRINT_ON_CLOSE, }
{ PROTECTION, }
{ READ_CHECK, }
{ REVISION, }
{ SEQUENTIAL_ONLY, }
{ SUBMIT_ON_CLOSE, }
{ SUPERSEDE, }
{ TEMPORARY, }
{ TRUNCATE_ON_CLOSE, }
{ USER_FILE_OPEN, }
{ WINDOW_SIZE, }
{ WRITE_CHECK, }
```

```
{ AFTER_IMAGE, }
{ AFTER_NAME, }
{ AUDIT-TRAIL, }
{ AUDIT-NAME, }
{ BEFORE_IMAGE, }
{ BEFORE_NAME, }
{ RECOVERY_UNIT, }
```

```
{ CHANGES, }
{ DATA_AREA, }
{ DATA_FILL, }
{ DATA_KEY_COMPRESSION, }
{ DATA_RECORD_COMPRESSION, }
{ DUPLICATES, }
{ INDEX_AREA, }
{ INDEX_COMPRESSION, }
{ INDEX_FILL, }
{ LEVEL_INDEX_AREA, }
{ NAMES, }
{ NULL_KEY, }
{ NULL_VALUE, }
{ PROLOGUE, }
{ SEG_LENGTH, }
{ SEG_POSITION, }
{ SEG_TYPE, }
```

```
{ BLOCK_SPAN, }
```

Source Listing

N 8
16-Sep-1984 00:42:36
15-Sep-1984 22:43:40VAX-11 Pascal V2.4-277
_82558DUA28:EDF.SRCJEDFVALUE.PAS;1 (2) Page 26

```
1328      (FALSE,FALSE,TRUE,FALSE),      { CARRIAGE CONTROL }
1329      (FALSE,TRUE,FALSE,FALSE),      { CONTROL_FIELD_SIZE, }
1330      (FALSE,FALSE,TRUE,FALSE),      { FORMAT }
1331      (FALSE,TRUE,FALSE,FALSE),      { SIZE }
1332
1333      ( *
1334      KEY:      STR, NUM, QUAL, SW
1335      - )
1336      ( SHARING PRIMARY )
1337
1338      (FALSE,FALSE,FALSE,TRUE),      { DELETE }
1339      (FALSE,FALSE,FALSE,TRUE),      { GET }
1340      (FALSE,FALSE,FALSE,TRUE),      { MULTISTREAM }
1341      (FALSE,FALSE,FALSE,TRUE),      { PROHIBIT }
1342      (FALSE,FALSE,FALSE,TRUE),      { PUT }
1343      (FALSE,FALSE,FALSE,TRUE),      { UPDATE }
1344      (FALSE,FALSE,FALSE,TRUE),      { USER_INTERLOCK }
1345
1346      ( SYSTEM PRIMARY )
1347
1348      (TRUE,FALSE,FALSE,FALSE),      { DEVICE, }
1349      (FALSE,FALSE,TRUE,FALSE),      { SOURCE, }
1350      (FALSE,FALSE,TRUE,FALSE),      { TARGET, }
1351
1352      );
```



```
1355 { *
1356 This is the QTAB array, which controls the asking and processing of questions.
1357 - }
1358 QTAB := (
1359 { *
1360 QUESTION OFFSET
1361 ANSWER_CLASS,      DEFAULT_OK,      DEFAULT,      LOW_BOUND,      HIGH_BOUND,      KEY_TABLE,      STATE_TABLE
1362 - }
1363 { EDFSK_DATA_FILE_NAME }
1364 (STRING_ANSWER,      TRUE,      0,      0,      0,      0,      0),
1365 { EDFSK_FDL_TITLE }
1366 (STRING_ANSWER,      TRUE,      0,      0,      0,      0,      0),
1367 { EDFSK_KEY_NAME }
1368 (STRING_ANSWER,      TRUE,      0,      0,      0,      0,      0),
1369 { EDFSK_ANALYSIS }
1370 (STRING_ANSWER,      TRUE,      0,      0,      0,      0,      0),
1371 { EDFSK_OUTPUT }
1372 (STRING_ANSWER,      TRUE,      0,      0,      0,      0,      0),
1373 { EDFSK_DATA_KEY_COMP }
1374 (REAL_ANSWER,      TRUE,      0,      -99,      99,      0,      0),
1375 { EDFSK_DATA_RECORD_COMP }
1376 (REAL_ANSWER,      TRUE,      0,      -99,      99,      0,      0),
1377 { EDFSK_INDEX_RECORD_COMP }
1378 (REAL_ANSWER,      TRUE,      0,      -99,      99,      0,      0),
1379 { *
1380 QUESTION OFFSET
1381 ANSWER_CLASS,      DEFAULT_OK,      DEFAULT,      LOW_BOUND,      HIGH_BOUND,      KEY_TABLE,      STATE_TABLE
1382 - }
1383 { EDFSK_KEY_COMP_WANTED }
1384 (BOOLEAN_ANSWER,      TRUE,      EDFSK_YES,      0,      0,      0,      0),
1385 { EDFSK_REC_COMP_WANTED }
1386 (BOOLEAN_ANSWER,      TRUE,      EDFSK_YES,      0,      0,      0,      0),
1387 { EDFSK_IDX_COMP_WANTED }
1388 (BOOLEAN_ANSWER,      TRUE,      EDFSK_YES,      0,      0,      0,      0),
1389 { EDFSK_ASCENDING_ADDED }
1390 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1391 { EDFSK_ASCENDING_LOAD }
1392 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1393 { EDFSK_BLOCK_SPAN }
1394 (BOOLEAN_ANSWER,      TRUE,      EDFSK_YES,      0,      0,      0,      0),
1395 { EDFSK_CONFIRM }
1396 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1397 { EDFSK_SEGMENTED }
1398 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1399 { EDFSK_GLOBAL_WANTED }
1400 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1401 { *
1402 QUESTION OFFSET
1403 ANSWER_CLASS,      DEFAULT_OK,      DEFAULT,      LOW_BOUND,      HIGH_BOUND,      KEY_TABLE,      STATE_TABLE
1404 - }
1405 { EDFSK_KEY_CHANGES }
1406 (BOOLEAN_ANSWER,      TRUE,      EDFSK_YES,      0,      0,      0,      0),
1407 { EDFSK_KEY_DIST }
1408 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1409 { EDFSK_KEY_DUPS }
1410 (BOOLEAN_ANSWER,      TRUE,      EDFSK_NO,      0,      0,      0,      0),
1411
```

1412	(EDFSK RETURN)						
1413	(BOOLEAN ANSWER,	TRUE,	0,	0,	0,	0,	0),
1414	(EDFSK CLUSTER_SIZE)						
1415	(INTEGER ANSWER,	TRUE,	3,	1,	EDFSC_1GIGA,	0,	0),
1416	(EDFSK ACTIVE KEY)						
1417	(INTEGER ANSWER,	TRUE,	0,	0,	0,	0,	0),
1418	(+						
1419	QUESTION OFFSET						
1420	ANSWER_CLASS,	DEFAULT_OK,	DEFAULT,	LOW_BOUND,	HIGH_BOUND,	KEY_TABLE,	STATE_TABLE
1421	(-						
1422	(EDFSK ADDED COUNT)						
1423	(INTEGER ANSWER,	TRUE,	0,	0,	EDFSC_1GIGA,	0,	0),
1424	(EDFSK ADDED COUNT_LOW)						
1425	(INTEGER ANSWER,	TRUE,	0,	0,	EDFSC_1GIGA,	0,	0),
1426	(EDFSK ADDED COUNT_HIGH)						
1427	(INTEGER ANSWER,	TRUE,	100000,	0,	EDFSC_1GIGA,	0,	0),
1428	(EDFSK BLOCKS IN_BUCKET)						
1429	(INTEGER ANSWER,	TRUE,	32,	1,	BKTSC_MAXBKTSIZ,	0,	0),
1430	(EDFSK BUCKET WEIGHT)						
1431	(KEYWORD ANSWER,	TRUE,	EDFSC_FLATTER_FILES,	0,	0,	0,	0),
1432	(EDFSK CARR CTAL)						
1433	(KEYWORD ANSWER,	TRUE,	FDLSC_CR,	0,	0,	0,	0),
1434	(EDFSK CONTROL_SIZE)						
1435	(INTEGER ANSWER,	TRUE,	2,	1,	255,	0,	0),
1436	(+						
1437	QUESTION OFFSET						
1438	ANSWER_CLASS,	DEFAULT_OK,	DEFAULT,	LOW_BOUND,	HIGH_BOUND,	KEY_TABLE,	STATE_TABLE
1439	(-						
1440	(EDFSK CURRENT_FUNCTION)						
1441	(KEYWORD ANSWER,	TRUE,	EDFSC_HELP,	0,	0,	0,	0),
1442	(EDFSK DESIGN CYCLE)						
1443	(KEYWORD ANSWER,	TRUE,	EDFSC_WP,	0,	0,	0,	0),
1444	(EDFSK DESIRED_FILL)						
1445	(INTEGER ANSWER,	TRUE,	100,	0,	100,	0,	0),
1446	(EDFSK FILL LOW)						
1447	(INTEGER ANSWER,	TRUE,	50,	0,	100,	0,	0),
1448	(EDFSK FILL HIGH)						
1449	(INTEGER ANSWER,	TRUE,	100,	0,	100,	0,	0),
1450	(+						
1451	QUESTION OFFSET						
1452	ANSWER_CLASS,	DEFAULT_OK,	DEFAULT,	LOW_BOUND,	HIGH_BOUND,	KEY_TABLE,	STATE_TABLE
1453	(-						
1454	(EDFSK GLOBAL COUNT)						
1455	(INTEGER ANSWER,	FALSE,	0,	0,	65535,	0,	0),
1456	(EDFSK GRANULARITY)						
1457	(KEYWORD ANSWER,	TRUE,	EDFSC_THREE,	0,	0,	0,	0),
1458	(EDFSK INITIAL COUNT)						
1459	(INTEGER ANSWER,	FALSE,	0,	0,	EDFSC_1GIGA,	0,	0),
1460	(EDFSK INITIAL COUNT_LOW)						
1461	(INTEGER ANSWER,	TRUE,	0,	0,	EDFSC_1GIGA,	0,	0),
1462	(EDFSK INITIAL COUNT_HIGH)						
1463	(INTEGER ANSWER,	TRUE,	100000,	0,	EDFSC_1GIGA,	0,	0),
1464	(EDFSK KEY POSITION)						
1465	(INTEGER ANSWER,	TRUE,	0,	0,	EDFSC_MAXRECSIZ,	0,	0),
1466	(EDFSK KEY LOW)						
1467	(INTEGER ANSWER,	TRUE,	1,	0,	0,	0,	0),
1468	(EDFSK KEY_HIGH)						

```
1469 (INTEGER ANSWER TRUE, 255, 0, 0, 0, 0),
1470 ( EDISK KEY SIZE )
1471 (INTEGER ANSWER, FALSE, 0, 0, 0, 0),
1472 ( +
1473 QUESTION OFFSET
1474 ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1475 - )
1476 ( EDISK KEY TYPE )
1477 (KEYWORD ANSWER TRUE, FDLSC_STG, 0, 0, 0, 0),
1478 ( EDISK LOAD METHOD )
1479 (KEYWORD ANSWER TRUE, EDISK_FAST_CONVERT, 0, 0, 0, 0),
1480 ( EDISK MAX RECORD_SIZE )
1481 (INTEGER ANSWER FALSE, 0, 0, 0, 0),
1482 ( EDISK MEAN RECORD_SIZE )
1483 (INTEGER ANSWER FALSE, 0, 1, EDISK_MAXRECSIZ, 0, 0),
1484 ( EDISK NUMBER DUPS )
1485 (INTEGER ANSWER TRUE, 0, 0, EDISK_1GIGA, 0, 0),
1486 ( EDISK NUMBER KEYS )
1487 (INTEGER ANSWER TRUE, 1, 1, 255, 0, 0),
1488 ( +
1489 QUESTION OFFSET
1490 ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1491 - )
1492 ( EDISK NUMBER RECORDS )
1493 (INTEGER ANSWER FALSE, 0, 0, EDISK_1GIGA, 0, 0),
1494 ( EDISK PROLOGUE_VERSION )
1495 (INTEGER ANSWER TRUE, 3, 0, 3, 0, 0),
1496 ( EDISK PROMPTING )
1497 (KEYWORD ANSWER TRUE, EDISK_FULL, 0, 0, 0, 0),
1498 ( EDISK RECORD FORMAT )
1499 (KEYWORD ANSWER TRUE, FDLSC_VAR, 0, 0, 0, 0),
1500 ( EDISK RESPONSES )
1501 (KEYWORD ANSWER TRUE, EDISK_AUTO, 0, 0, 0, 0),
1502 ( EDISK SCRIPT OPTION )
1503 (KEYWORD ANSWER FALSE, 0, 0, 0, 0, 0),
1504 ( EDISK SET FUNCTION )
1505 (KEYWORD ANSWER FALSE, 0, 0, 0, 0, 0),
1506 ( EDISK SIZE LOW )
1507 (INTEGER ANSWER TRUE, 1, 1, EDISK_MAXRECSIZ, 0, 0),
1508 ( EDISK SIZE HIGH )
1509 (INTEGER ANSWER TRUE, 1000, 1, EDISK_MAXRECSIZ, 0, 0),
1510 ( EDISK SURFACE_OPTION )
1511 (KEYWORD ANSWER TRUE, EDISK_LINE_SURFACE, 0, 0, 0, 0),
1512 ( +
1513 QUESTION OFFSET
1514 ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1515 - )
1516 ( EDISK TEST PRIMARY )
1517 (KEYWORD ANSWER TRUE, FDLSC_FILE, 0, 0, 0, 0),
1518 ( EDISK TEST SECOND/ Y )
1519 (OBJECT ANSWER FALSE, 0, 0, 0, 0, 0),
1520 ( EDISK TEST SECONDARY_VALUE )
1521 (OBJECT ANSWER FALSE, 0, 0, 0, 0, 0),
1522 );
1523
1524 NULL_STRING := (
1525
```

```
1526  
1527  
1528      0,          ( DSC$W_LENGTH )  
1529      DSC$K_DTYPE_T, ( DSC$B_DTYPE )  
1530      DSC$K_CLASS_D, ( DSC$B_CLASS )  
1531      NIL          ( DSC$A_POINTER )  
1532  
1533      );  
1534  
1535      LINE_OBJECT_TEMPLATE := (  
1536          SEC,          ( LINE_OBJECT_TYPE )  
1537          NIL,          ( FORE )  
1538          NIL,          ( BACK )  
1539  
1540          (             ( COMMENT )  
1541          0,  
1542          DSC$K_DTYPE_T,  
1543          DSC$K_CLASS_D,  
1544          NIL  
1545          ),  
1546  
1547          (             ( STRING )  
1548          0,  
1549          DSC$K_DTYPE_T,  
1550          DSC$K_CLASS_D,  
1551          NIL  
1552          ),  
1553  
1554          KEY,          ( PRIMARY )  
1555          0,            ( PRINUM )  
1556          DUMMY_SECONDARY$, ( SECONDARY )  
1557          0,            ( SECNUM )  
1558          0,            ( QUALIFIER )  
1559          0,            ( NUMBER )  
1560          TRUE,         ( SWITCH )  
1561          0,            ( OWNER_UIC )  
1562  
1563          (             ( PROT_MASK )  
1564          FALSE,  
1565          FALSE,  
1566          FALSE,  
1567          FALSE,  
1568          FALSE,  
1569          FALSE,  
1570          FALSE,  
1571          FALSE,  
1572          FALSE,  
1573          FALSE,  
1574          FALSE,  
1575          FALSE,  
1576          FALSE,  
1577          FALSE,  
1578          FALSE,  
1579          FALSE,  
1580          FALSE,  
1581          FALSE,  
1582          FALSE,
```


EDFVAR
V04-000

Source Listing

M 8
16-Sep-1984 00:42:36
15-Sep-1984 22:43:40

VAX-11 Pascal V2.4-277
_S255\$DUA28:[EDF.SRC]EDFVALUE.PAS;1 (3) Page 31

```
1583 1 FALSE,  
1584 1 FALSE,  
1585 1 FALSE,  
1586 1 FALSE,  
1587 1 FALSE,  
1588 1 FALSE,  
1589 1 FALSE,  
1590 1 FALSE,  
1591 1 FALSE,  
1592 1 FALSE,  
1593 1 FALSE,  
1594 1 FALSE,  
1595 1 FALSE,  
1596 1 ),  
1597 1  
1598 1 0, { FID1 }  
1599 1 0, { FID2 }  
1600 1 0 { FID3 }  
1601 1  
1602 1 );  
1603 1  
1604 1 ( End of file SRC$:EDFVALUE.PAS )  
1605  
1606 END.  
1607 ( End of file: SRC$:EDFVAR.PAS )
```

```
00 00 00 54 53 45 44 5F 4C 44 46 08
00000001 00000041 00000000
00000001 00000041 00000000
00 54 54 02
00000001 00000041 00000000
45 4C 49 46 5F 4C 41 4E 52 55 4F 4A 0C
```

```
00000000 00000000' 00000000 00000010'
```

```
00000000 00000018' 00000000 00000024'
```

```
00000000 00000028' 00000000 00000038'
```

```
00
```

```
00
```

```
00
```

```
00
```

```
00
```

```
00
```

```
.TITLE EDFVAR
.IDENT \V04-000\

00000 .PSECT $CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2

00000 C.AAA: .LONG 0,65,1
0000C .ASCII <8>\FDL_DEST\<0><0><0>
00018 C.AAB: .LONG 0,65,1
00024 .ASCII <2>\TT\<0>
00028 C.AAC: .LONG 0,65,1
00034 .ASCII <12>\JOURNAL_FILE\

00000 .PSECT $LOCAL,-
00000 PIC,CON,REL,LCL,NOSHR,NOEXE,RD,WRT,2

00000 FDL_DEST:
00000 .ADDRESS .+16,0,C.AAA,0
00010 .BLKB 1
00011 3
00014 TT: .ADDRESS .+16,0,C.AAB,0
00024 .BLKB 1
00025 3
00028 JOURNAL_FILE:
00028 .ADDRESS .+16,0,C.AAC,0
00038 .BLKB 1
00039 3
0003C JOURNAL_ENABLED:
0003C .BYTE 0
0003D .BLKB 3
00040 JOURNAL_FILENAME:
00040 .BLKB 255
0013F 1
00140 INPUT_FILENAME_DESC:
00140 .BLKB 8
00148 OUTPUT_FILENAME_DESC:
00148 .BLKB 8
00150 ANALYSIS_FILENAME_DESC:
00150 .BLKB 8
00158 RES_OUTPUT_FILENAME_DESC:
00158 .BLKB 8
00160 DEFAULT_FILENAME_DESC:
00160 .BLKB 8
00168 NL_DEV_DESC:
00168 .BLKB 8
00170 ANALYSIS_SPECIFIED:
00170 .BYTE 0
00171 .BLKB 3
00174 ANALYSIS_ONLY:
00174 .BYTE 0
00175 .BLKB 3
00178 EDITING: .BYTE 0
00179 .BLKB 3
0017C TAKE_DEFAULTS:
0017C .BYTE 0
0017D .BLKB 3
00180 NO_INPUT:
00180 .BYTE 0
```

Generated Code

16-Sep-1984 00:42:36
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277 Page
DISK\$VMSMASTER:[EDF.SRC]EDFVAR.PAS:1 (4)

Page 33

00 3A 54 55 50 54 55 4F 24 53 59

```

00181          .BLKB      3
00184 AUTO_TUNE:
00184          .BYTE      0
00185          .BLKB      3
00188 SYSS$OUTPUT NAME:
00188          .ASCII      \SYSS$OUTPUT:\<0>
00194 SYSS$INPUT ERROR:
00194          .BYTE      0
00195          .BLKB      3
00198 RMS_INPUT ERROR:
00198          .BYTE      0
00199          .BLKB      3
0019C RMS_OUTPUT ERROR:
0019C          .BYTE      0
0019D          .BLKB      3
001A0 CONTROL_ZEE TYPED:
001A0          .BYTE      0
001A1          .BLKB      3
001A4 MAIN_CTRLZ:
001A4          .BYTE      0
001A5          .BLKB      3
001A8 MAIN_LEVEL:
001A8          .BYTE      1
001A9          .BLKB      3
001AC QUESTION_TYPED:
001AC          .BYTE      0
001AD          .BLKB      3
001B0 XY_PLOT: .BLKL      416
00B30 COCOR_PLOT:
00B30          .BLKL      416
00EB0 XY_ARRAY:
00EB0          .BLKL      819
01B7C COLOR_ARRAY:
01B7C          .BLKL      819
02848 COLOR_ROW:
02848          .BLKB      63
02887          1
02888 BREAKPOINT LEFT:
02888          .BCKL      1
0288C BREAKPOINT MID:
0288C          .BCKL      1
02890 BREAKPOINT RIGHT:
02890          .BCKL      1
02894 DEPTHPOINT LEFT:
02894          .BCKL      1
02898 DEPTHPOINT MID:
02898          .BCKL      1
0289C DEPTHPOINT RIGHT:
0289C          .BCKL      1
028A0 EXAMPOINT LEFT:
028A0          .BLKL      1
028A4 EXAMPOINT MID:
028A4          .BLKL      1
028A8 EXAMPOINT RIGHT:
028A8          .BLKL      1
028AC NUMPOINT LEFT:
028AC          .BLKL      1

```

EDF
V04
COM P
/
/
/
/
/
/
COM P
I
S
S
T
F
P
C
N
C
F
T
COM C
E
P
C

COM
CER

07	06	03	07	05	04	04	07	04	0F	10	03	06	00
												05	06
11	17	14	09	0F	05	06	08	09	03	03	06	08	00
0A	13	0E	0F	11	10	13	14	0A	11	13	05	09	13
0B	0C	0B	07	0B	08	0C	06	0B	09	11	0A	0B	13
0F	0A	12	06	11	10	10	0D	0C	0B	09	10	11	10
0B	06	0C	0F	09	10	0E	13	09	0F	13	0F	0E	0E
0F	0F	0F	0E	0C	09	0A	07	0C	0B	13	0A	0B	0A
11	13	08	04	10	11	0F	0E	0D	0A	13	0D	13	09
0B	0B	0E	11	09	09	0F	0F	0B	0A	0A	0E	05	0C
0A	0A	17	14	09	09	07	0D	0B	0C	0A	0B	0A	0B
06	12	10	0A	00	00	00	06	0A	0B	04	11	0A	11
			06	06	06	0E	06	03	0B	0B	03	06	04

00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
0000003F	00000000	3B9AC9FF	00000000	00000000
0000FFFF	00FFFFFF	3B9AC9FF	00000000	00000000
00000000	3B9AC9FF	3B9AC9FF	00000000	00000000
00000000	00000000	000000FF	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	000000FF	000000FF
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
0000003F	00000000	3B9AC9FF	00000000	00000000
00000000	00000000	00000000	3B9AC9FF	3B9AC9FF
3B9AC9FF	00000000	00000000	00000000	00000000
00000000	00000000	00000000	0000FFFC	00007FFF
00000000	00000000	3B9AC9FF	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	0000FFFF	00000000	00000000	00000000

028B0 NUMPOINT_MID:
028B0 .BLKL 1
028B4 NUMPOINT_RIGHT:
028B4 .BLKL 1
028B8 PAGEPOINT_LEFT:
028B8 .BLKL 1
028BC PAGEPOINT_MID:
028BC .BLKL 1
028C0 PAGEPOINT_RIGHT:
028C0 .BLKL 1
028C4 GRAPH_TYPE:
028C4 .BLKL 1
028C8 CURRENT_GRAPH_INDEX:
028C8 .BLKL 1
028CC LAST_GRAPH_INDEX:
028CC .BLKL 1
028D0 STEPS: .BLKL 1
028D4 Y_LABEL: .BLKB 32
028F4 PRIMARY_WIDTH:
028F4 .SIGNED_BYTE 0,6,3,16,15,4,7,4,4,5,7,3,6,7,6,5
02902
02904 SECONDARY_WIDTH:
02904 .SIGNED_BYTE 0,8,6,3,3,9,8,6,5,15,9,20,23,17,19,9,5,19,-
02912 17,10,20,19,16,17,15,14,19,10,19,11,10,17,-
02920 9,8,6,12,8,11,7,11,12,11,16,17,16,9,11,12,-
0292E 13,16,16,17,6,18,10,15,14,15,19,15,9,-
0293C 19,14,16,9,15,12,6,8,10,8,10,19,11,12,7,-
0294A 10,9,12,14,15,15,15,6,19,13,19,10,13,14,-
02958 15,17,16,4,8,19,17,12,5,14,10,10,8,15,15,-
02966 9,9,17,14,11,11,11,10,11,10,12,11,13,7,9,-
02974 9,20,23,10,10,17,10,17,4,8,10,6,0,0,0,10,-
02982 16,18,6,4,6,3,11,8,3,6,14,6,6,6
02990
02998 .BLKB 1
0299C SECONDARY_MAX:
0299C .LONG 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-
029B0 0,0,0,0,0,0,99999999,0,63,0,0,99999999,-
029C4 16777215,65535,0,0,99999999,99999999,0,-
029D8 0,0,255,0,0,0,0,0,0,0,255,255,0,0,0,0,-
029EC 255,0,0,0,0,0,0,0,0,0,0,0,0,99999999,-
02A00 0,63,99999999,99999999,0,0,0,0,0,0,-
02A14 99999999,32767,65532,0,0,0,0,0,99999999,-
02A28 0,0,0,0,0,0,0,0,0,0,65535,0,0,0,0
02A3C
02A50
02A64
02A78
02ABC
02AA0
02AB4
02AC8
02ADC
02AF0
02B04
02B18
02B2C
02B40

EDFVAR
V04-000

Generated Code

00000000	00000000	3B9AC9FF	00000000	00000000									
00000000	00000000	00000000	00000000	00000000									
00000063	00000064	000000FE	00000000	00000000									
00000064	00000063	000000FE	00000000	00000063									
00000003	000000FF	00000000	00000000	000000FE									
00000000	00000000	00000000	00003FAB	000000FF									
00000000	00000000	00007DF0	00000000	000000FF									
00000000	00000000	00000000	00000000	00000000									
00000000	00000000	00000000	00000000	00000000									
09	07	0C	06	01	03	04	0A	0B	0E	0D	04	08	0F
												00	02

6D 30 5B 1B

6D 31 5B 1B

6D 34 5B 1B

6D 35 5B 1B

6D 37 5B 1B

00 00 00 00

00 00 22 22

00 00 00 09

00 00 00 00 00 09 0A 0D

00

09

1B

27

17

1A

3F

07

0 9
16-Sep-1984 00:42:36
3-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277
DISK&VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (4)

Page 35

02B48 .LONG 0,0,99999999,0,0,0,0,0,0,0,0,254,100,-
02B5C 99,99,0,254,99,100,254,0,0,255,3,255,-
02B70 16299,0,0,0,255,0,32240,0,0,0,0,0,0,0,0,-
02B84 0

02BF8 PRI_SEQ: .SIGNED_BYTE 15,8,4,13,14,11,10,4,3,1,6,12,7,9,2,0

02C06 ANSI_RESET: .ASCII <27>\[0m\
02C08 .ASCII <27>\[0m\
02C0C ANSI_BOLD: .ASCII <27>\[1m\
02C0C .ASCII <27>\[1m\
02C10 ANSI_UNDERSCORE: .ASCII <27>\[4m\
02C10 .ASCII <27>\[4m\
02C14 ANSI_BLINK: .ASCII <27>\[5m\
02C14 .ASCII <27>\[5m\
02C18 ANSI_REVERSE: .ASCII <27>\[7m\
02C18 .ASCII <27>\[7m\
02C1C VID_STRING4: .BLKB 4
02C1C .BLKB 4
02C20 NULL_STRING4: .ASCII <0><0><0><0>
02C20 .ASCII <0><0><0><0>
02C24 EMPTY_STRING: .ASCII \"'\<0><0>
02C24 .ASCII \"'\<0><0>
02C28 SHIFT: .ASCII <9><0><0><0>
02C28 .ASCII <9><0><0><0>
02C2C CRLF_SHIFT: .ASCII <13><10><9><0><0><0><0><0>
02C2C .ASCII <13><10><9><0><0><0><0><0>
02C34 LOW_SHIFT: .BLKB 3
02C34 .BLKB 3
02C37 1
02C38 NULL_CHAR: .BYTE 0
02C38 .BLKB 3
02C39 .BLKB 3
02C3C TAB: .BYTE 9
02C3C .BLKB 3
02C40 ESCAPE: .BYTE 27
02C40 .BLKB 3
02C44 APOSTROPHE: .BYTE ^A\'
02C44 .BLKB 3
02C45 .BLKB 3
02C48 CONTROL_W: .BYTE 23
02C48 .BLKB 3
02C49 .BLKB 3
02C4C CONTROL_Z: .BYTE 26
02C4C .BLKB 3
02C4D .BLKB 3
02C50 QUESTION_MARK: .BYTE ^A\?
02C50 .BLKB 3
02C51 .BLKB 3
02C54 ERR_CHAR: .BLKB 1
02C54 .BLKB 3
02C55 .BLKB 3
02C58 CONTROL_G: .BYTE 7
02C58 .BLKB 7

00 00 0A 0D

00

00

00

00

00000001

00000000

00000050

00000018

00

01

00

```
02C59 .BLKB 3
02C5C CRLF: .ASCII <13><10><0><0>
02C60 TERMINAL_TYPE:
02C60 .BLKL 1
02C64 TERMINAL_SPEED:
02C64 .BLKL 1
02C68 ANSI_CRT:
02C68 .BYTE 0
02C69 .BLKB 3
02C6C REGIS: .BYTE 0
02C6D .BLKB 3
02C70 DEC_CRT: .BYTE 0
02C71 .BLKB 3
02C74 DEV_TYPE:
02C74 .BLKL 1
02C78 VIDEO_TERMINAL:
02C78 .BYTE 0
02C79 .BLKB 3
02C7C VID_TERM:
02C7C .BLKL 1
02C80 SCREEN_FLAGS:
02C80 .BLKB 20
02C94 OUT_LINE:
02C94 .BLKB 257
02D95
02D98 ONE: .LONG 1
02D9C CHFFLAGS:
02D9C .LONG 0
02DA0 FLAGS: .BLKB 1
02DA1 3
02DA4 TEMP_FDL3TYPE:
02DA4 .BLKB 3
02DA7 1
02DAB LINE_WIDTH:
02DAB .BLKL 1
02DAC LINES_PER_PAGE:
02DAC .BLKL 1
02DB0 DEST_IS_TERMINAL:
02DB0 .BLKB 1
02DB1 3
02DB4 LINES_SHOWN:
02DB4 .BLKL 1
02DB8 MINIMUM_TERM_WIDTH:
02DB8 .LONG 80
02DBC MINIMUM_VIDEO_PAGE:
02DBC .LONG 24
02DC0 SCROLLING_SET:
02DC0 .BYTE 0
02DC1 .BLKB 3
02DC4 FULL_PROMPT:
02DC4 .BYTE 1
02DC5 .BLKB 3
02DC8 TEMP_FULL_PROMPT:
02DC8 .BYTE 0
02DC9 .BLKB 3
02DCC ORIG_TIME:
02DCC .BLKF 1
```

```
08
00000000
00000001
00000001
00000011
00000017

74 74 41 20 79 72 61 64 6E 6F 63 65 53 20
   00 00 20 73 65 74 75 62 69 72
           00 00 50 4C 48 46 44 45
20 20 20 20 20 20 20 20 20 20 20 20 20 20
31 20 58 41 56 20 20 20 20 20 20 20 20 20
   72 6F 74 69 64 45 20 4C 44 46 20 31
           00000028

20 72 6F 66 20 22 3F 22 20 65 70 79 54 28
72 6F 77 79 65 48 20 66 6F 20 74 73 69 6C
           00 29 73 64
64 45 20 4C 44 46 20 31 31 20 58 41 56 20
   00 20 72 6F 74 69
20 4E 52 55 54 45 52 20 73 73 65 72 50 20
5E 28 20 65 75 6E 69 74 6E 6F 63 20 6F 74
6E 65 4D 20 6E 69 61 4D 20 72 6F 66 20 5A
           00 00 00 20 29 75
```

```
00000000
00000000
00000000
00000000
00000000
```

```
02DD0 QUAD_TIME:
02DD0 .BLKB 8
02DD8 DEFAULT_PRIMARY:
02DD8 .BYTE 8
02DD9 .BLKB 3
02DDC DEFAULT_PRIMUM:
02DDC .LONG 0
02DE0 COL_ONE: .LONG 1
02DE4 LINE_ONE: .LONG 1
02DE8 LOWER_LINE: .LONG 17
02DE8 .LONG 17
02DEC PROMPT_LINE: .LONG 23
02DEC .LONG 23
02DF0 PARAM_BLOCK: .BLKB 36
02DF0 .BLKB 36
02E14 SEC_ATTR:
02E14 .ASCII \ Secondary Attributes \<0><0>
02E22 EDFHLP_STRING:
02E2C .ASCII \EDFHLP\<0><0>
02E34 IDENT_STRING:
02E34 .ASCII \ VAX-11 FDL Editor\
02E42
02E50 IDENT_STRING_LENGTH:
02E5C .LONG 40
02E60 QUES_HINT:
02E60 .ASCII \ (Type "?" for list of Keywords)\<0>
02E6E
02E7C
02E80 EDF_HEADER:
02E80 .ASCII \ VAX-11 FDL Editor \<0>
02E8E
02E94 CONTINUE_TEXT:
02E94 .ASCII \ Press RETURN to continue (^Z for Main M\
02EA2 .ASCII \enu) \<0><0><0>
02EB0
02EBE
02EC4 ISTATUS: .BLKL 1
02EC8 FAB_DUMMY:
02EC8 .BLKA 1
02ECC RAB_DUMMY:
02ECC .BLKA 1
02ED0 FDL_BLOCK:
02ED0 .BLKA 1
02ED4 DEF_CURRENT:
02ED4 .LONG 0
02ED8 DEF_SCRATCH:
02ED8 .LONG 0
02EDC DEF_HEAD:
02EDC .LONG 0
02EE0 DEF_TAIL:
02EE0 .LONG 0
02EE4 DEF_SUCC:
02EE4 .LONG 0
02EE8 DEF_PRED:
```

00000000	02EE8	.LONG	0
00000000	02EE8	DEF_ANL_HEAD:	
00000000	02EE8	.LONG	0
00000000	02EE8	DEF_ANL_TAIL:	
00000000	02EE8	.LONG	0
00000000	02EE8	DEF_SAVE_HEAD:	
00000000	02EE8	.LONG	0
00000000	02EE8	DEF_SAVE_TAIL:	
00000000	02EE8	.LONG	0
01	02EFC	POINTING_AT_DEFINITION:	
	02EFC	.BYTE	1
	02EFD	.BLKB	3
00	02F00	FILE_CREATED:	
	02F00	.BYTE	0
	02F01	.BLKB	3
	02F04	INPUT_DESC:	
	02F04	.BLKB	8
	02F0C	INPUT_STRING:	
	02F0C	.BLKB	255
	03008		1
	0300C	INPUT_VALUE:	
	0300C	.BLKL	1
	03010	INPUT_NUMBER:	
	03010	.BLKL	1
	03014	QUAD_DESC:	
	03014	.BLKB	9
	0301D		3
	03020	LINKED: .BLKB	1
	03021		3
	03024	ACTIVE_AREA:	
	03024	.BLKL	1
	03028	ACTIVE_PRIMARY:	
	03028	.BLKB	1
	03029		3
	0302C	VARIABLE_RECORDS:	
	0302C	.BLKB	1
	0302D		3
	03030	CUR_MAX_REC:	
	03030	.BLKL	1
	03034	BYTES_PER_BUCKET:	
	03034	.BLKL	1
	03038	BUCKET_DEFAULT:	
	03038	.BLKL	1
	0303C	PRIMARY_INDEX_BUCKETS:	
	0303C	.BLKL	1
	03040	INIT_PRIMARY_BUCKETS:	
	03040	.BLKB	32
	030C0	ADDED_PRIMARY_BUCKETS:	
	030C0	.BLKL	32
	03140	INIT_NUMBER_BUCKETS:	
	03140	.BLKL	32
	031C0	ADDED_NUMBER_BUCKETS:	
	031C0	.BLKB	32
	03240	RECS_PER_BUCKET:	
	03240	.BLKL	32
	032C0	DEEPEST: .BLKL	1
	032C4	FIRST_PLOT:	

				00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000063	FFFFFF9D	00000000 00000001 01
00000000	00000000	00000063	FFFFFF9D	00000000 00000001 01
00000000	00000000	00000063	FFFFFF9D	00000000 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002

032C4	.BLKB	1
032C5		3
032C8	OPTIMIZING:	
032C8	.BLKB	1
032C9		3
032CC	VISIBLE_QUESTION:	
032CC	.BLKB	1
032CD		3
032D0	WAIT_HELP:	
032D0	.BLKB	1
032D1		3
032D4	QTAB:	
032D8	.LONG	0
032D8	.BYTE	1
032D9	.LONG	0,0,0,0,0,0
032ED		
032F1	.BYTE	1
032F2	.LONG	0,0,0,0,0,0
03306		
0330A	.BYTE	1
0330B	.LONG	0,0,0,0,0,0
0331F		
03323	.BYTE	1
03324	.LONG	0,0,0,0,0,0
03338		
0333C	.BYTE	1
0333D	.LONG	0,0,0,0,0,1
03351		
03355	.BYTE	1
03356	.LONG	0,-99,99,0,0,1
0336A		
0336E	.BYTE	1
0336F	.LONG	0,-99,99,0,0,1
03383		
03387	.BYTE	1
03388	.LONG	0,-99,99,0,0,2
0339C		
033A0	.BYTE	1
033A1	.LONG	1,0,0,0,0,2
033B5		
033B9	.BYTE	1
033BA	.LONG	1,0,0,0,0,2
033CE		
033D2	.BYTE	1
033D3	.LONG	1,0,0,0,0,2
033E7		
033EB	.BYTE	1
033EC	.LONG	0,0,0,0,0,2
03400		
03404	.BYTE	1
03405	.LONG	0,0,0,0,0,2
03419		
0341D	.BYTE	1
0341E	.LONG	1,0,0,0,0,2
03432		
03436	.BYTE	1
03437	.LONG	0,0,0,0,0,2
0344B		

Sym
CL I
CL I
CL I
CL I
CL I
CL I
CL I
CL I
CL I
CL I
CL I
CL I
EDT
EDT
EDT
EDT
EDT
EDT
EDT
EDT
EDT
LIB
LIB
STR
STR
SYS

00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000003 01
00000000	00000000	3B9AC9FF	00000001	00000003 00000003 01
00000000	00000000	00000000	00000000	00000000 00000003 01
00000000	00000000	3B9AC9FF	00000000	00000000 00000003 01
00000000	00000000	3B9AC9FF	00000000	00000000 00000003 01
00000000	00000000	3B9AC9FF	00000000	000186A0 00000003 01
00000000	00000000	0000003F	00000001	00000020 00000004 01
00000000	00000000	00000000	00000000	00000001 00000004 01
00000000	00000000	00000000	00000000	00000009 00000003 01
00000000	00000000	000000FF	00000001	00000002 00000004 01
00000000	00000000	00000000	00000000	00000003 00000004 01
00000000	00000000	00000000	00000000	00000003 00000003 01
00000000	00000000	00000064	00000000	00000064 00000003 01
00000000	00000000	00000064	00000000	00000032 00000003

0344F	.BYTE	1
03450	.LONG	0,0,0,0,0,2
03464		
03468	.BYTE	1
03469	.LONG	0,0,0,0,0,2
0347D		
03481	.BYTE	1
03482	.LONG	1,0,0,0,0,2
03496		
0349A	.BYTE	1
0349B	.LONG	0,0,0,0,0,2
034AF		
034B3	.BYTE	1
034B4	.LONG	0,0,0,0,0,2
034C8		
034CC	.BYTE	1
034CD	.LONG	0,0,0,0,0,3
034E1		
034E5	.BYTE	1
034E6	.LONG	3,1,999999999,0,0,3
034FA		
034FE	.BYTE	1
034FF	.LONG	0,0,0,0,0,3
03513		
03517	.BYTE	1
03518	.LONG	0,0,999999999,0,0,3
0352C		
03530	.BYTE	1
03531	.LONG	0,0,999999999,0,0,3
03545		
03549	.BYTE	1
0354A	.LONG	100000,0,999999999,0,0,3
0355E		
03562	.BYTE	1
03563	.LONG	32,1,63,0,0,4
03577		
0357B	.BYTE	1
0357C	.LONG	1,0,0,0,0,4
03590		
03594	.BYTE	1
03595	.LONG	9,0,0,0,0,3
035A9		
035AD	.BYTE	1
035AE	.LONG	2,1,255,0,0,4
035C2		
035C6	.BYTE	1
035C7	.LONG	3,0,0,0,0,4
035DB		
035DF	.BYTE	1
035E0	.LONG	3,0,0,0,0,3
035F4		
035F8	.BYTE	1
035F9	.LONG	100,0,100,0,0,3
0360D		
03611	.BYTE	1
03612	.LONG	50,0,100,0,0,3
03626		

[illegible]

EDFVAR
V04-000

Generated Code

J 9
16-Sep-1984 00:42:36
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277
DISK\$VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (4)

Page 41

00000000	00000000	00000064	00000000	0000000001	0362A	.BYTE	1
				0000000003	0362B	.LONG	100,0,100,0,0,3
				0000000000	0363F		
00000000	00000000	0000FFFF	00000000	0000000001	03643	.BYTE	0
				0000000004	03644	.LONG	0,0,65535,0,0,4
				0000000001	03658		
00000000	00000000	00000000	00000000	0000000002	0365C	.BYTE	1
				0000000003	0365D	.LONG	2,0,0,0,0,3
				0000000000	03671		
00000000	00000000	3B9AC9FF	00000000	0000000003	03675	.BYTE	0
				0000000003	03676	.LONG	0,0,999999999,0,0,3
				0000000001	0368A		
00000000	00000000	3B9AC9FF	00000000	0000000000	0368E	.BYTE	1
				0000000003	0368F	.LONG	0,0,999999999,0,0,3
				0000000001	036A3		
00000000	00000000	3B9AC9FF	00000000	000186A0	036A7	.BYTE	1
				0000000003	036A8	.LONG	100000,0,999999999,0,0,3
				0000000001	036BC		
00000000	00000000	00007DF0	00000000	0000000000	036C0	.BYTE	1
				0000000003	036C1	.LONG	0,0,32240,0,0,3
				0000000001	036D5		
00000000	00000000	00000000	00000000	0000000001	036D9	.BYTE	1
				0000000003	036DA	.LONG	1,0,0,0,0,3
				0000000001	036EE		
00000000	00000000	00000000	00000000	000000FF	036F2	.BYTE	1
				0000000003	036F3	.LONG	255,0,0,0,0,3
				0000000000	03707		
00000000	00000000	00000000	00000000	0000000000	0370B	.BYTE	0
				0000000004	0370C	.LONG	0,0,0,0,0,4
				0000000001	03720		
00000000	00000000	00000000	00000000	0000000021	03724	.BYTE	1
				0000000004	03725	.LONG	33,0,0,0,0,4
				0000000001	03739		
00000000	00000000	00000000	00000000	0000000000	0373D	.BYTE	1
				0000000003	0373E	.LONG	0,0,0,0,0,3
				0000000000	03752		
00000000	00000000	00000000	00000000	0000000000	03756	.BYTE	0
				0000000003	03757	.LONG	0,0,0,0,0,3
				0000000000	0376B		
00000000	00000000	00007DF0	00000001	0000000000	0376F	.BYTE	0
				0000000003	03770	.LONG	0,1,32240,0,0,3
				0000000001	03784		
00000000	00000000	3B9AC9FF	00000000	0000000000	03788	.BYTE	1
				0000000003	03789	.LONG	0,0,999999999,0,0,3
				0000000001	0379D		
00000000	00000000	000000FF	00000001	0000000001	037A1	.BYTE	1
				0000000003	037A2	.LONG	1,1,255,0,0,3
				0000000000	037B6		
00000000	00000000	3B9AC9FF	00000000	0000000000	037BA	.BYTE	0
				0000000003	037BB	.LONG	0,0,999999999,0,0,3
				0000000001	037CF		
00000000	00000000	00000003	00000000	0000000003	037D3	.BYTE	1
				0000000004	037D4	.LONG	3,0,3,0,0,4
				0000000001	037E8		
00000000	00000000	00000000	00000000	0000000000	037EC	.BYTE	1
				0000000004	037ED	.LONG	0,0,0,0,0,4
					03801		

_82

Vir
Sta
Ima
Ima
Num
Num
Num
Num
Use
Num
Ima
Map
Est

Per

Tot
Usi
Tot

Num

12

A t

LIN
LIB
EXE

[illegible]

03805	.BYTE	1
03806	.LONG	14,0,0,0,0,4
0381A		
0381E	.BYTE	1
0381F	.LONG	0,0,0,0,0,4
03833		
03837	.BYTE	0
03838	.LONG	0,0,0,0,0,4
0384C		
03850	.BYTE	0
03851	.LONG	0,0,0,0,0,3
03865		
03869	.BYTE	1
0386A	.LONG	1,1,32240,0,0,3
0387E		
03882	.BYTE	1
03883	.LONG	1000,1,32240,0,0,4
03897		
0389B	.BYTE	1
0389C	.LONG	5,0,0,0,0,4
038B0		
038B4	.BYTE	1
038B5	.LONG	8,0,0,0,0,6
038C9		
038CD	.BYTE	0
038CE	.LONG	0,0,0,0,0,6
038E2		
038E6	.BYTE	0
038E7	.LONG	0,0,0,0,0
038FB	.BLKB	1
038FC	QTAB_OFFSET:	
038FC	.BLKL	1
03900	SDATA:	.SIGNED_WORD 0
03902		.SIGNED_BYTE 14,2
03904		.LONG 0
03908		.SIGNED_WORD 0
0390A		.SIGNED_BYTE 14,2
0390C		.LONG 0
03910		.SIGNED_WORD 0
03912		.SIGNED_BYTE 14,2
03914		.LONG 0
03918		.SIGNED_WORD 0
0391A		.SIGNED_BYTE 14,2
0391C		.LONG 0
03920		.SIGNED_WORD 0
03922		.SIGNED_BYTE 14,2
03924		.LONG 0
03928		.SIGNED_WORD 0
0392A		.SIGNED_BYTE 14,2
0392C		.LONG 0
03930	RDATA:	.BLKF 11
0395C	BDATA:	.BYTE 0,-
0396A		0,0,0,0
03975		.BLKB 3
03978	IDATA:	.BLKL 74
03AA0	VDATA:	.BYTE 0,-
03AAE		0,-

00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00

```
03ABC      0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0  
03ACA  
03AD8  
03ADF          .BLKB   1  
03AE0 TEMP_STRING255:  
03AE0          .BLKB  255  
03BDF          1  
03BE0 TEMP_DESCRIPTOR:  
03BE0          .BLKB   8  
03BE8 QUERY_FLAG:  
03BE8          .BLKB   1  
03BE9          3  
03BEC LOW_KEY:.LONG    0  
03BF0 HIGH_KEY:  
03BF0          .LONG    0  
03BF4 LOW_AREA:  
03BF4          .LONG    0  
03BF8 HIGH_AREA:  
03BF8          .LONG    0  
03BFC FOUND AREA:  
03BFC          .BYTE    0  
03bfd        .BLKB     3  
03C00 FOUND_KEY:  
03C00          .BYTE    0  
03C01          .BLKB     3  
03C04 FOUND_0:.BYTE    0  
03C05          .BLKB     3  
03C08 MAX_KEY_SIZE:  
03C08          .BLKL     1  
03C0c MIN_KEY_SIZE:  
03C0c          .BLKL     1  
03C10 SEGMENT_WANTED:  
03C10          .BYTE    0,0,0,0,0,0,0,0  
03C18 SEGMENT_POSITION:  
03C18          .BLKL     8  
03C38 SEGMENT_LENGTH:  
03C38          .BLKL     8  
03C58 SEGMENT_NUMBER:  
03C58          .BLKL     1  
03C5c BUCKET_OVERHEAD:  
03C5c          .BLKL     1  
03C60 MIN_BUCKET:  
03C60          .BLKL     1  
03C64 ENTRY_SIZE:  
03C64          .BLKL     1  
03C68 LOWMAX: .BLKL     1  
03C6c EXTRA:  .BLKL     1  
03C70 CUR_MAX_FIXED:  
03C70          .BLKL     1  
03C74 MAX_STRING_ANSWER_LENGTH:  
03C74          .BLKL     1  
03C78 OLD_COUNT:  
03C78          .BLKL     1  
03C7c GLOBAL_SET:  
03C7c          .BLKB     1  
03C7d          3  
03C80 NUMBER_KEYS_SET:
```

EXE

Mod

EDT

EDT

EDT
EDT

EDT
EDT

EDT

EDT

EDT
EDTEDT
FDT

EDT

EDT

EDIT
EDIT

EDT
EDT

EDT

EDT
EDTEDT
EDT

EDT

EDT

EDIT
EDIT

EDT
EDT

EDT

EDT
EDT

EDIT
EDIT

EDT
EDT

EDT

EDY
EDY

EDIT
EDIT

EDT

EDT

EDY
EDY

FD1

ED1

ED1

ED1
ED1

ED1

ED1

ED1
ED1ED1
ED1

ED1

ED1

1

00	03C80	.BYTE	0
	03C81	.BLKB	3
	03C84	ISAM_ORG:	
	03C84	.BLKB	1
	03C85		3
	03C88	MAX_KEY_POSITION:	
	03C88	.BLKL	1
	03C8C	TEMP_REAL:	
	03C8C	.BLKF	1
	03C90	TEMP_STATUS:	
	03C90	.BLKL	1
	03C94	TEMP_INT2:	
	03C94	.BLKL	1
	03C98	DEF:	.BLKL 1
	03C9C	NULL_STRING:	
0000	03C9C	.SIGNED_WORD	0
02 0E	03C9E	.SIGNED_BYTE	14,2
00000000	03CA0	.LONG	0
	03CA4	TEST:	.BLKB 64
	03CE4	FULL_CHOICE:	
	03CE4	.BLKB	1
	03CE5		3
	03CE8	SEC_TYPE:	
88888880	03CE8	.FIELD	1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03CEC		1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,-
00000001	03CEC		1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03CF0		1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF0		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
00000000	03CF0		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF4		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
88282000	03CF4		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF8		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF8		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:0,1:0,-
82288202	03CF8		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
	03CFC		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03CFC		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
88888288	03CFC		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,-
	03D00		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
	03D00		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
88882288	03D00		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
	03D04		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
	03D04		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
88888828	03D04		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
	03D08		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:1,1:0,-
	03D08		1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
11118888	03D0C		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
	03D0C		1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,-
18822282	03D0C		1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,-
	03D10		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:0,1:0,-
	03D10		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
82228888	03D10		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:1,1:0,1:0,1:0,-
	03D14		1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,1:0,-
	03D14		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:0,1:0,-

```
81828808 03D14 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
03D18 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
03D18 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,-
28080488 03D18 1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
03D18 1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:0,-
03D1C 1:0,1:1,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,-
82888888 03D1C 1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:0,-
03D1C 1:0,1:1,1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,-
03D20 1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
84181818 03D20 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
03D20 1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
03D24 1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:1,-
28288822 03D24 1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:1,1:0,-
03D24 1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
03D28 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
42220812 03D28 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
03D28 1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,-
03D2C 1:0,1:0,1:0,1:1,1:0,1:0,1:1,1:0,1:0,1:0,-
03D2C 1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
88824248 03D2C 1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
03D30 1:1,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,-
03D30 1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
04418888 03D30 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
03D34 1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
03D34 1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,-
03D34 1:0,1:0,1:1,1:0,4:0
03D34 LINE_OBJECT TEMPLATE:
03D34 .BYTE 1
00000000 00000000 03D35 .LONG 0,0
0000 03D3D .SIGNED_WORD 0
02 0E 03D3F .SIGNED_BYTE 14,2
00000000 03D41 .LONG 0
0000 03D45 .SIGNED_WORD 0
02 0E 03D47 .SIGNED_BYTE 14,2
00000000 03D49 .LONG 0
0B 03D4D .BYTE 11
00000000 03D4E .LONG 0
00 03D52 .BYTE 0
00000000 00000000 03D53 .LONG 0,0,0
01 03D5F .BYTE 1
00000000 03D60 .LONG 0
00000000 03D64 .FIELD 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
03D68 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
03D68 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
03D68 1:0,1:0
00000000 00000000 00000000 03D68 .LONG 0,0,0
03D74 .END
```


COMMAND QUALIFIERS

PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS\$:EDFVAR/OBJ=OBJ\$:EDFVAR MSRC\$:EDFVAR

/CHECK=(NOBOUNDS, NOCASE_SELECTORS, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)

/DEBUG=(NOSYMBOLS, NOTRACEBACK)

/ENVIRONMENT= \$255\$DUA28:[EDF.OBJ]EDFVAR.PEN;1

/LIST= \$255\$DUA28:[EDF.LIS]EDFVAR.LIS;1

/OBJECT= \$255\$DUA28:[EDF.OBJ]EDFVAR.OBJ;1

/NOCROSS_REFERENCE /ERROR_LIMIT=30 /NOG_FLOATING /MACHINE_CODE /NOOLD_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	97	00:00.4	00:03.8
Source Analysis	732	00:14.5	02:40.1
Source Listing	41	00:02.3	00:05.3
Tree Construction	0	00:00.0	00:00.0
Flow Analysis	0	00:00.0	00:00.0
Profit Analysis	0	00:00.0	00:00.0
Context Analysis	0	00:00.0	00:00.0
Name Packing	0	00:00.0	00:00.0
Code Selection	0	00:00.0	00:00.0
Final	106	00:03.3	00:08.2
TOTAL	976	00:20.5	02:57.4

COMPILATION STATISTICS

CPU Time: 00:20.5 (4706 Lines/Minute)
Elapsed Time: 02:57.4
Page Faults: 976
Compilation Complete

0129 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY